



# **STIGH – STIGmergia de Hiperligações**

**Artur Manuel Sancho Marques**  
**(Licenciado)**

**Dissertação para a obtenção do Grau de Mestre em Comércio Electrónico e Internet**

**Orientador: Doutor Ivo Dias de Sousa**

**Setembro de 2006**

## **Resumo**

Stigmergia – do Grego *stigma* (marca) e *ergon* (trabalho) – significa «a marca do trabalho».

Na natureza abundam soluções stigmérgicas, como os caminhos otimizados por formigas, marcados com feromonas.

Neste trabalho engenha-se uma nova forma de hiperligação, para hipertexto: a hiperligação stigmérgica. A ideia é transformar os utilizadores de uma página, na WWW, em formigas, que deixam a sua marca, ao fazerem click numa ligação. Esse click é considerado uma manifestação de preferências.

Com a utilização, a comunidade das hiperligações stigmérgicas presentes na página, sem controlo externo, trata de eliminar os destinos negligenciados, preservar os destinos queridos, e fazer emergir novas ligações potencialmente apreciadas.

Chama-se a este sistema STIGH: STIGmergia de Hiperligações.

O funcionamento do STIGH é transparente. Um visitante da página não deverá distinguir as ligações stigmérgicas de hiperligações comuns, em HTML. Isto significa que o sistema tem potencial para fazer a captura implícita das preferências de navegação da comunidade dos seus utilizadores.

Este documento disserta sobre o STIGH, da ideia subjacente, à sua concretização na forma de um componente, para websites.

## **Palavras-chave**

Hiperligação, Hipertexto, Stigmergia, Web, Conceção, Algoritmos.

## **Abstract**

Stigmergy – from the Greek *stigma* (mark) and *ergon* (work) – means “the mark of the work”.

In nature, stigmergic solutions abound, like the optimized paths ants travel, marked with their pheromones.

In this work, a new form of hyperlink, for hypertext, is engineered: the stigmergic hyperlink. The idea is to transform WWW users in ants, who leave their mark, when clicking a link. The click is considered a demonstration of preferences.

As the page is visited and clicked, its community of stigmergic links will extinguish neglected destinations, preserve favored URLs, and make emerge new, potentially appreciated, links.

This system is called STIGH: STIGmergic Hyperlinks.

The STIGH works transparently. The page visitor shouldn't distinguish between the stigmergic and the common HTML links. This means that the system has the potential to implicitly capture the navigation preferences of its users.

This document is a dissertation about the STIGH, from the original idea to the real thing, in the form of a website component.

## **Keywords**

Hyperlink, Hypertext, Stigmergy, Web, Design, Algorithms.

# Índice de conteúdos

Resumo-----	2
Palavras-chave -----	2
Abstract -----	3
Keywords -----	3
Índice de conteúdos-----	4
Índice de imagens-----	7
Índice de tabelas -----	8
Índice de tabelas -----	8
1. Stigmergia -----	9
1. Stigmergia -----	9
1.1. Conceito -----	9
1.2. Motivação para a stigmergia de hiperligações -----	11
2. Sistema proposto: STIGH -----	13
2.1. Visão geral-----	13
2.2. Detalhes de funcionamento -----	14
2.3. Substituição de ligações -----	14
3. Enquadramento científico e filosófico-----	17
3.1. Localização científica, objectivo e ética -----	17
3.2. Enquadramento filosófico-----	18
3.3. A metodologia (processo de investigação) -----	20
4. Modelação -----	25
5. Hipertexto e hiperligações-----	29
5.1. Origens e evolução-----	29
5.2. Hipertexto doméstico e hipertexto selvagem-----	31
5.3. Outros sistemas de hiperligações-----	33
6. Utilização (ilustrada) do STIGH-----	35
6.1. Início -----	36
6.2. Após um primeiro click-----	39
6.3. Após um segundo click-----	43
6.4. Após um terceiro click-----	45
6.5. Após um quarto click -----	47
6.6. Quinto e sexto clicks-----	48
6.7. Sétimo e oitavo clicks-----	50

6.8. Ao nono click, antes de extinções-----	52
6.9. Décimo click: a faísca para uma extinção tripla -----	53
6.10. Cálculo de probabilidades e roleta proporcional -----	54
6.11. Resumo -----	55
7. Ficheiros de registo de estado-----	58
8. Observações -----	61
8.1. Página com um cStigh: STIGH(1) -----	61
8.2. Página com dois cStigh: STIGH(2)-----	61
8.3. Páginas STIGH(n), n>2-----	62
8.4. Satisfação do objectivo -----	63
9. Engenharia do sistema -----	64
9.1. Classes e outras estruturas de informação -----	64
9.2. Declaração de hiperligações stigmérgicas -----	66
9.3. Prefixo declarativo -----	68
9.4. Ficheiro de configuração: stigh.cfg -----	69
9.5. Classe cStigh – Discussão, a partir das constantes -----	70
9.5.1. Apresentação da informação de estado: stighInfo-----	71
9.5.2. Discussão, baseada nos membros de dados protected -----	74
9.5.3. Preservação da informação de estado: writeStateToFile -----	77
9.5.4. Propriedades públicas -----	79
9.5.5. Propriedades protected -----	80
9.5.6. Métodos públicos -----	81
9.5.7. Métodos protected -----	83
9.5.8. Métodos herdados, especializados-----	84
9.5.9. Métodos para o desenho de um cStigh, na página-----	84
9.5.10. Métodos relacionados com o ciclo de vida do objecto, em .NET -----	85
9.5.11. Resposta ao evento de click: OnClick -----	85
9.5.12. Métodos de suporte ao funcionamento de cada hiperligação stigmérgica-----	86
9.5.13. A extinção: sparkExtinctions + extinguish-----	87
9.6. Classes auxiliares -----	90
9.6.1. Classe ClassAmRandom -----	90
9.6.2. Classe ClassAmUtil-----	92
9.6.3. Classe cStighDemo -----	95
10. Sumário e conclusão -----	97
10.1. Sumário-----	97

10.2. Limitações -----	97
10.3. Trabalho Futuro -----	98
10.3.1. Melhorias na usabilidade de hiperligações-----	98
10.3.2. Aprofundamento do conhecimento da stigmergia de hiperligações-----	100
10.3.3. Possibilidade no sentido de Inteligência Artificial em objectos de hipertexto-----	101
10.3.4. Possibilidade no sentido de métricas mais refinadas, de utilização de websites -----	101
11. Anexos -----	103
11.1. Versão de demonstração: <a href="http://stigh.org">http://stigh.org</a> -----	103
11.2. Galeria de diagramas de fluxo: <a href="http://stigh.org/album/fd">http://stigh.org/album/fd</a> -----	103
11.3. Galeria de diagramas de sequência: <a href="http://stigh.org/album/sd">http://stigh.org/album/sd</a> -----	103
11.4. Código fonte: <a href="http://stigh.org/sc">http://stigh.org/sc</a> -----	103
Referências -----	104

## Índice de imagens

Figura 1 – Metodologia no paradigma DR. Adaptado de (Vaishnavi & Kuechler 2004) -----	23
Figura 2 – Diagrama de casos de utilização-----	25
Figura 3 – Diagrama de classes-----	27
Figura 4 – <i>websitesasgraphs</i> para a página de demonstração do STIGH, em <a href="http://stigh.org">http://stigh.org</a> -----	32
Figura 5 – Exemplo ilustrado de utilização do STIGH - inicialização-----	36
Figura 6 – Exemplo ilustrado de utilização do STIGH - sequência do primeiro click-----	38
Figura 7 – Exemplo ilustrado de utilização do STIGH - um click ocorrido-----	40
Figura 8 – Exemplo ilustrado de utilização do STIGH - sequência do segundo click -----	42
Figura 9 – Exemplo ilustrado de utilização do STIGH - dois clicks ocorridos -----	44
Figura 10 – Exemplo ilustrado de utilização do STIGH - três clicks ocorridos -----	46
Figura 11 – Exemplo ilustrado de utilização do STIGH - quatro clicks ocorridos-----	47
Figura 12 – Exemplo ilustrado de utilização do STIGH - seis clicks ocorridos -----	49
Figura 13 – Exemplo ilustrado de utilização do STIGH - oito clicks ocorridos-----	51
Figura 14 – Exemplo ilustrado de utilização do STIGH - imediatamente antes de extinções -----	52
Figura 15 – Roleta proporcional a ser utilizada nas extinções, ao décimo click-----	54
Figura 16 – Exemplo ilustrado de utilização do STIGH - dez clicks (e três extinções) ocorridos---	57
Figura 17 – cStigh – sparkExtinctions – Diagrama de fluxo -----	63
Figura 18 – Class View, no ambiente Visual Studio, para a solução desenvolvida-----	65
Figura 19 – Estrutura tpStighState, para o estado interno de cada objecto cStigh -----	65
Figura 20 – Ferramentas para a inserção de ligações stigmérgicas em páginas -----	66
Figura 21 – Ficheiro de configuração stigh.cfg -----	69
Figura 22 – Informação sobre a ligação stigmérgica em <i>tooltip</i> e na <i>statusbar</i> -----	69
Figura 23 – Classe cStigh (representação parcial do seu diagrama de classes)-----	70
Figura 24 – Exemplo de caixa de diálogo para confirmação da navegação-----	71
Figura 25 – cStigh - <i>Fields</i> / membros de dados -----	74
Figura 26 – cStigh - <i>Properties</i> / propriedades -----	79
Figura 27 – cStigh – Métodos-----	81
Figura 28 – cStigh – OnClick – Diagrama de fluxo -----	89
Figura 29 – ClassAmRandom-----	90
Figura 30 – ClassAmUtil -----	92
Figura 31 – cStighDemo -----	95

## Índice de tabelas

Tabela 1 – Objectivos e métodos .....	17
Tabela 2 – Paradigmas quantitativo e qualitativo. Adaptado de (Creswell 2002).....	20
Tabela 3 – Paradigmas quantitativo e DR.....	22
Tabela 4 – Passos da investigação e seus resultados .....	24
Tabela 5 – Cálculo de probabilidades para a roleta proporcional.....	54
Tabela 6 – Consulta online dos registos de evolução de estado .....	58

# 1. Stigmergia

## 1.1. Conceito

A origem etimológica de «stigmergia» está nas palavras gregas *stigma* (marca) e *ergon* (trabalho). O conceito foi introduzido pelo biólogo Francês Pierre-Paul Grassé, na sequência dos seus trabalhos sobre insectos sociais (Grassé 1977), onde se observam comportamentos colectivos cuja «inteligência» parece transcender a soma das inteligências dos indivíduos que os protagonizam.

Grassé observou que as térmitas constroem estruturas complexíssimas, à sua escala, não porque sejam dirigidas por alguma inteligência central, mas antes como resultado da interacção de cada indivíduo com o seu espaço imediato, que pode estimulá-lo mais ou menos, para a edificação de flocos de terra, consoante a concentração de saliva nesses aglomerados (a «marca para o trabalho»). Ou seja, nenhuma térmita necessita de ter uma memória global do território; basta-lhe a memória suficiente para uma tarefa específica, que por sua vez só depende do ambiente imediato.

Antes de Grassé, embora sem identificar a forma de comunicação «invisível» das térmitas, o Sul-Africano Eugène Marais dedicou um livro à descrição do funcionamento de uma termiteira (Marais 1969), fazendo analogias com o funcionamento do corpo humano.

Outros exemplos de insectos sociais encontram-se nas abelhas, nas formigas e nas vespas.

Admitindo «emergência» como a formação de padrões de comportamentos complexos, a partir de unidades básicas e sectoriais dos mesmos, os insectos sociais podem ser considerados agentes de «emergência».

Um exemplo mais próximo de emergência é o próprio cérebro humano, uma vez que, isoladamente, cada neurónio pode ser visto como uma relativamente simples bomba de sódio/potássio (Czerner 2002); todavia, o colectivo dessas células parece capaz de funções extraordinariamente mais elevadas...

Sob certa perspectiva, todos os organismos vivos e todas as sociedades manifestam fenómenos de emergência, estudados em áreas de investigação como «Inteligência Colectiva», «Inteligência de Enxame» ou «Comportamentos Emergentes». No caso particular da célula biológica, enquanto sistema que se auto-mantém e que constitui o «tijolo» de organizações sucessivamente mais complexas, como tecidos/orgãos/seres, está-se perante uma manifestação de *autopoiesis* (Humberto R. Maturana 1992) – do Grego *auto* (própria) e *poiesis* (criação) – expressão introduzida pelos biólogos Francisco Varela e Humberto Maturana.

O interesse da aplicação de stigmergia a problemas computacionais, está na possibilidade de aplicação de algoritmos relativamente simples a espaços de dados que serão sucessivamente transformados, por oposição a algoritmos relativamente complexos, actuando sobre (grandes) espaços de dados estáticos.

Por exemplo, alguns problemas genéricos em Inteligência Artificial, como a identificação do caminho mais curto entre dois pontos e outros relacionados com a Teoria dos Grafos, podem ser abordados, tomando as formigas como inspiração (*Ant-based algorithms*). Jean-Louis Deneubourg estudou a capacidade de formigas para encontrarem o caminho mais curto para uma fonte de alimentos (Camazine et al. 2003). A comunicação indirecta das formigas faz-se por feromonas cuja presença vai sendo reforçada no caminho mais curto; assim, mesmo que de início existam n trajectos alternativos, ao cabo de algum tempo o caminho mais curto – por ser percorrido com maior facilidade – estará mais marcado por feromonas (maior «stigma»), o que vai atraindo mais e mais formigas a percorrê-lo (maior «ergon») – isto é stigmergia.

O conceito de stigmergia está assim relacionado com sistemas auto-organizados. Um sistema auto-organizado é aquele onde emerge ordem, inteiramente como consequência das interacções entre os seus componentes individuais.

## 1.2. Motivação para a stigmergia de hiperligações

Os motores da procura na Internet, como os disponibilizados por [Google.com](http://Google.com), [Yahoo.com](http://Yahoo.com), [Accoona.com](http://Accoona.com) e outras empresas, são preciosos para o estabelecimento de caminhos entre «consumidores» e «produtores» de conteúdos; mas, entre outros factores condicionantes da sua eficácia, conta-se o estatismo das hiperligações, na tecnologia de hipertexto dominante (HTML). Este estatismo permite, por exemplo, a sobrevivência de ligações, muito após o desaparecimento dos recursos de destino aos quais ligavam – «ligações mortas».

Para lá da potencial desactualização dos conteúdos indexados perante a realidade, só solúvel aquando das actualizações periódicas e massivas dos índices mantidos, o serviço prestado pelos grandes motores de procura, ignora conteúdos que, pela forma como são servidos, não são procuráveis de maneira completamente automática; por exemplo: conteúdos dinâmicos, gerados em função de parâmetros, conteúdos protegidos por senha, ou conteúdos protegidos por mecanismos anti-*bots*.

Também conteúdos de nicho tendem a ser esquecidos, por ser baixa, ou mesmo nula, a sua interligação com outras páginas.

Os problemas relacionados com a desproporção extrema, entre (poucos) motores de procura e (muitos) conteúdos – problemas de centralização – podem, por natureza, ser minimizados, reduzindo significativamente a desproporção: descentralizando.

Os problemas relacionados com o estatismo das hiperligações, como a preservação de «ligações mortas» – problemas de (falta) de utilidade – podem, por natureza, ser minimizados com uma dinâmica que valore a utilidade das ligações.

Assumindo que uma ligação é tanto mais útil, para o seu universo de utilizadores, quanto mais vezes for seleccionada, será de esperar que «ligações mortas» tendam para a utilidade nula.

Esta investigação pretende conceber um sistema de hiperligações descentralizado (pelo universo dos utilizadores de uma página) e dinâmico (mudando a certo ritmo de utilizações), com inspiração stigmergica: um sistema que permita capturar as preferências de navegação dos utilizadores (as utilidades das hiperligações) e actuar automaticamente, extinguindo/substituindo os links preteridos (menos úteis), e preservando os links preferidos (mais úteis).

Neste sistema, os visitantes desempenharão o papel equivalente ao de formigas, largando «feromonas», quando seguirem certo trajecto; isto é, quando seleccionarem certa hiperligação, estarão a marcá-la como mais útil, relativamente às alternativas. Ao fazerem-no, estarão, indirectamente, a contribuir para a «evaporação» de trajectos alternativos; ou seja, a manifestar preferência por todas as outras hiperligações.

O objectivo é que este seja um sistema auto-organizado de hiperligações – um contributo para uma Internet auto-organizada (Sousa 2005).

Este documento disserta sobre um sistema **STIG**mérgico de **H**iperligações (STIGH).

## 2. Sistema proposto: STIGH

### 2.1. Visão geral

Cada página, que contenha pelo menos uma «hiperligação stigmérgica», é um sistema stigmérgico de hiperligações: um STIGH.

Cada hiperligação stigmérgica parece-se com uma ligação HTML, mas distingue-se por ter associada uma vitalidade, ou vida, que é reforçada sempre que é seleccionada, e que se degrada com uma certa periodicidade, medida em número de eventos de selecção sobre ligações stigmérgicas, na mesma página.

Destes reforços e degradações, resultam a anulação da vida de alguns *links* e o reforço da vida de outros. Destes fenómenos emerge a «sobrevivência» dos recursos stigmérgicos efectivamente utilizados pelos visitantes, e o desaparecimento/substituição dos recursos negligenciados

Quando a intensidade de alguma ligação se anula, essa ligação será substituída por outra, escolhida aleatoriamente, na página de destino de alguma das ligações stigmérgicas «vivas». Qual a ligação stigmérgica da qual se vai extrair, aleatoriamente, uma ligação substituta, é algo que se determina probabilisticamente, por um sorteio em roleta proporcional, que concede maior possibilidade aos destinos com maior vitalidade.

O sistema é orientado à simplicidade: cada evento de selecção de uma hiperligação só provoca a actualização da sua intensidade e, periodicamente, o envelhecimento de toda a página, desencadeando eventuais episódios de extinções/substituições.

Não há lugar à comunicação entre páginas, daí a selecção aleatória (sem diálogo) de um sucessor, na página eleita para originá-lo.

Este conhecimento limitado do meio reflecte o exemplo dos insectos sociais e cria condições para que seja genuinamente stigmérgico o comportamento do sistema, em caso de uso generalizado. É previsível que a utilidade para os utilizadores cresça com a generalização do STIGH, pois as ligações «vivas» reflectirão cada vez mais as preferências dos visitantes, cujas escolhas as foram justificando.

## 2.2. Detalhes de funcionamento

Seja  $\{L_1, L_2, \dots, L_n\}$  um conjunto de hiperligações stigmérgicas, com intensidades  $\{i_1, i_2, \dots, i_n\}$ .

Estas intensidades estão limitadas a um limite superior (**iMaxLife**), de forma a estabelecer-se um tecto para a influência do passado distante no presente: este tecto contribui para evitar que «aberrações» na utilização passada tenham um efeito duradouro.

A cada **iTrigger** eventos de selecção, envelhecem-se as intensidades  $i$  de todos os  $n$  links, perante uma comunicação (**allGetWeaker**) do objecto que detectou que era altura de fazê-lo. O valor da degradação não é imposto: o sistema pode ser configurado quanto aos valores de recompensa e de penalização.

Na concretização corrente do STIGH, utilizam-se os valores:

- **iTrigger**  $\beta$  2
- recompensa  $\beta$  2 \* **iTrigger**
- penalização  $\beta$  recompensa / 2

Nesta configuração, o «envelhecimento» acontece a cada 2 clicks. Cada recompensa incrementa em 4 pontos a vitalidade da ligação seleccionada. Cada penalização custa 2 pontos a cada uma das ligações stigmérgicas na página.

### Qual o valor de $i$ para um novo link?

Um valor «neutro» será **iMaxLife**/2. Esta é a opção seguida na versão corrente.

Um valor que queira conceder maiores hipóteses de continuidade à ligação, deverá ser maior do que **iMaxLife**/2; e um valor que queira dificultar a subsistência do *hyperlink*, deverá atribuir-lhe uma intensidade inicial inferior a **iMaxLife**/2.

## 2.3. Substituição de ligações

Um autor pode editar páginas com qualquer número de ligações stigmérgicas, mas uma vez publicada a página, esse número permanecerá fixo: assim, quando uma ligação se extinguir (por anulação da sua vida), terá que ser substituída por outra.

**Para determinar a ligação sucessora**, faz-se um sorteio probabilístico, entre as ligações «sobreviventes» na página. Quanto mais elevada for a intensidade duma ligação, maior será a sua probabilidade de eleição. Chama-se roleta proporcional a este tipo de sorteio.

Por exemplo, uma ligação L1 com  $i_1=Y$  tem o dobro da probabilidade de ser eleita, do que uma ligação L2, com  $i_2=Y/2$ . Sendo I a soma de todas as intensidades, a probabilidade P de uma ligação  $L_n$  ser a escolhida é  $P = i_n/I$ .

Num segundo momento, elege-se uma ligação qualquer, dentro da página sorteada pela roleta - é esse destino que substitui o destino extinto.

Computacionalmente, estes sorteios implicam a geração de números pseudo-aleatórios e o cálculo de probabilidades. A selecção de uma ligação na página de destino, também obriga ao processamento dessa página, para identificação das ligações possíveis.

Assim, a extinção é um evento computacionalmente exigente, relativamente às simplicíssimas somas e subtracções de vitalidade.

No contexto da extinção, levantam-se dois problemas interessantes: (1) o que fazer se a página de destino for um «beco sem saída»; isto é, sem conexões a outros URLs, logo sem possibilidade de gerar um link substituto?, e (2) o que fazer se o link substituto vier a constituir uma ligação da página a si própria?

Quanto a (1) repetir-se o processo, até conseguir o objectivo, não é uma verdadeira solução, porque a repetição até ao sucesso não é um algoritmo formal, por não garantir o término: afinal, pode sempre voltar a sortear-se o mesmo destino que é um «beco sem saída».

Cancelar-se a substituição do link cuja intensidade se tenha extinguido, é uma solução «fraca», por adiamento. Esta «solução» preservaria artificialmente uma hiperligação, já «morta», na expectativa de que o próximo sorteio para a sua substituição conduzisse a uma página «fértil», e não a um destino «estéril» de saídas.

Note-se que esta situação não tem exactamente os mesmos problemas da anterior, em que era impossível ter-se um algoritmo formal, pela insistência, porque o espaço de possibilidades era estático. Aqui, insistir-se-ia sobre um espaço de dados que poderia mudar, perante mais um evento de selecção.

A solução em vigor para (1) é um mecanismo de «mutação», que faz a substituição da ligação de intensidade nula por outra, seleccionada de forma radicalmente desconexa, partir de uma procura

aleatória, sobre as expressões mais procuradas na semana anterior, via [Google.com](http://Google.com). Chamou-se a este comportamento **randomGoogleZeitgeistUrl**. Detalhes da arquitectura do sistema são discutidos na secção 9.

As mutações são um mecanismo evolutivo, para fuga à estacionaridade, frequentemente transposto para a Informática, por exemplo em aplicações como algoritmos genéticos.

Quanto a (2) a «inutilidade» de uma página em ligar-se a si própria, deverá conduzir à sua extinção, *naturalmente*, por falta de interesse dos utilizadores. A solução em vigor é a de não rejeitar *a priori* estes destinos.

O STIGH tem potencial para promover a comunicação entre consumidores e fornecedores de conteúdos, na Internet. Em websites equipados com o sistema, as ligações disponibilizadas deverão reflectir as preferências dos utilizadores, ao longo do tempo; por outro lado, pela forma como a substituição de ligações está pensada, os *hyperlinks* externos dependem dos URLs aos quais os websites externos, com ou sem STIGH, liguem as suas páginas.

### 3. Enquadramento científico e filosófico

#### 3.1. Localização científica, objectivo e ética

Esta investigação situa-se na **área** de hipertexto, no **tópico** da *auto-organização de hiperligações, pela captura implícita das preferências de navegação de utilizadores num website, seguindo uma abordagem stigmérgica*.

A **questão** é: *consegue o sistema proposto (STIGH) manifestar a emergência de hiperligações, consonante com as preferências manifestadas pelos utilizadores da página?*

O **objectivo** cardinal é *averiguar se a abordagem stigmérgica proposta, é capaz, ou não, de conferir dinâmica às hiperligações de uma página, de maneira a que evoluam consoante as preferências de navegação dos visitantes, assumindo* que as preferências de navegação reflectem a utilidade das hiperligações e que a utilidade dum hiperligação é tanto maior, quanto mais vezes essa ligação for seleccionada.

Na Tabela 1, identificam-se objectivos «parcelares» e os métodos correspondentes.

Objectivo	Método(s)
Determinar o «estado do conhecimento», relativamente a hiperligações.  Determinar aplicações de stigmérgia em hipertexto.	- Revisão de literatura.
Construção do STIGH.	- Concepção e Desenvolvimento. <ul style="list-style-type: none"><li>• Modelação</li><li>• Especificações</li><li>• Algoritmos</li><li>• Programação</li></ul>
Aplicação do STIGH a um website.	- Experimentação. - Automação da experimentação.
Avaliação do funcionamento do STIGH.	- Mensuração do impacto.

Tabela 1 – Objectivos e métodos

Apesar da intencionada captura de preferências de navegação ser feita de forma transparente (sem que os utilizadores tenham consciência de que está a acontecer), uma vez que é preservado o anonimato do visitante, considera-se a investigação **ética**.

Note-se que são registados dados como o IP, um identificador de sessão, a data e hora de cada click, mas não há qualquer esforço de relacioná-los com outras informações ou de partilhá-los com outros sistemas; aliás, os dados mencionados poderiam estar omissos, sem que o funcionamento do STIGH fosse comprometido. O registo é feito apenas para facilitar a leitura dos eventos.

### 3.2. Enquadramento filosófico

Para esta investigação, consideraram-se inicialmente dois paradigmas: o **qualitativo** e o **quantitativo**.

Os paradigmas, assim como as teorias e os métodos de investigação que cada um segue, condicionam o desenvolvimento de um projecto. Conforme se fundamenta de seguida, esta investigação seguiu um paradigma de **Design** (*Design Research*), abordado de uma forma quantitativa. Há outras perspectivas e abordagens, situadas noutras esferas paradigmáticas, como a *Action Research and Design* na óptica dos autores Figueiredo e Cunha (António Dias de Figueiredo 2006).

**Ontologicamente** (em termos da natureza da realidade), serão as preferências dos utilizadores, em termos das hiperligações que seguem uma «realidade» singular, objectivamente mensurável, ou interpretável de múltiplas formas?

Pela assunção de quando um utilizador escolhe uma ligação, pretere todas as outras, então a preferência é uma realidade objectiva e mensurável: cada evento de escolha é uma afirmação das suas preferências de navegação.

Esta assunção é «natural», no sentido em que é legítimo esperar que, no geral, os utilizadores compreendam o seu papel de «actores» (Laurel 1993), num «palco» (o website) que reage consistentemente à acção de clicar em ligações.

Mesmo no cenário de utilizadores inexperientes, que interagem «à descoberta», até perceberem a lógica de navegação, ou no cenário de *bots*, como o *Googlebot*, que navegam para efeitos de indexação, pode argumentar-se que os primeiros serão residuais e/ou efémeros – pois o mero acto de navegarem até ao website inicia logo uma experiência (de aprendizagem) que deverá conduzi-los a um estado em que já não se justificará o rótulo de «inexperientes» –, e que os segundos são

residuais, relativamente ao universo de «Internautas», e que ao fazerem o que estiverem mecanizados para fazer, estarão a manifestar as suas preferências programadas.

Por outras palavras, considera-se a navegação um processo contínuo de manifestação de preferências. Assim, quanto ao objecto de estudo, o projecto começa por situar-se no paradigma quantitativo.

**Epistemologicamente** (quanto à natureza do conhecimento), as preferências de navegação num website, resultam directamente dos utilizadores.

A «morte»/substituição de uma hiperligação, reflecte a falta de preferência dos visitantes, e apenas deles (e não do investigador), pelo link em causa, relativamente a outros links. Neste sentido, a investigação alinha-se com o paradigma quantitativo.

As hiperligações são substituídas quando se lhes esgota a vitalidade, por inactividade, relativamente às outras hiperligações stigmérgicas. Uma vez que o valor inicial de vida de uma ligação substituta, é ditado pelo algoritmo, a dinâmica da substituição *depende* desse valor: um valor inicial elevado para a «vida» confere condições para uma maior longevidade aos links e menos mudanças na teia da página; um valor inicial baixo, confere condições para uma menor longevidade e maior «efervescência» na teia da página.

Todavia, apesar desta influência, o valor inicial da vida para uma ligação substituta, não afecta a captura de preferências em si mesma, a longo prazo. Assim, embora o investigador possa exercer uma influência seminal, não controla a utilização da página (o sistema stigmérgico), pelo que há um distanciamento positivista.

**Axiologicamente** (quanto aos valores) são relevantes, nesta investigação, o modelo, os algoritmos formais e o produto de software desenvolvidos.

Quanto à **retórica**, considerando que o sistema tem tradução em algoritmos formais, implementados numa linguagem artificial (C#), a investigação tem, nesse prisma, uma retórica sem ambiguidades.

Considera-se um algoritmo formal uma sequência finita de instruções, com um objectivo claro, com garantia de término, em que cada instrução reúne as características de (1) ter uma só interpretação possível (sem ambiguidade), (2) ser executável mecanicamente (sem requerer criatividade), (3) em tempo e com esforço finitos.

Considera-se uma linguagem artificial livre de contexto, aquela em que qualquer frase tem uma só interpretação possível, conferindo-lhe assim potencial para a escrita de algoritmos formais.

Todavia, é fácil de perceber que uma linguagem artificial não conduz necessariamente a algoritmos formais; por exemplo `main(){while(1);}` é um programa completo, em linguagem C, que resulta num ciclo infinito.

As frases em línguas naturais, como o português ou o inglês, dependem de um contexto, para a sua interpretação correcta, pelo que não são adequadas para a escrita de algoritmos formais.

Em termos comparativos, a Tabela 2 sintetiza as diferenças consideradas entre os paradigmas quantitativo e qualitativo. O paradigma quantitativo está **destacado**, devido aos pontos de contacto que foram identificados.

	<b>Paradigma Quantitativo</b>	<b>Paradigma Qualitativo</b>
Ontologia	Realidade objectiva.	Realidade subjectiva. São possíveis várias interpretações.
Epistemologia	Investigador distante e independente do que é investigado.	Investigador interage com o seu objecto de investigação.
Axiologia	Não se assumem tendências (de interpretação).	Tendências (de interpretação) assumidas.
Retórica	Linguagem formal.	Linguagem informal.

Tabela 2 – Paradigmas quantitativo e qualitativo. Adaptado de (Creswell 2002)

De seguida, discute-se a metodologia.

### **3.3. A metodologia (processo de investigação)**

É quanto à metodologia que o projecto STIGH não se identifica totalmente com o paradigma quantitativo.

O paradigma quantitativo está conotado com um processo de investigação que envolve actividades como observação, mensuração e análise estatística de dados. Este processo é considerado como lógica dedutiva, permitindo generalizações que podem ser validadas por métodos formais (Creswell 2002). Estas actividades são aplicáveis à investigação em curso.

A única dificuldade em enquadrar a investigação STIGH com o paradigma quantitativo, ao nível metodológico, está no facto de aqui *se construir um novo sistema*, logo algo sem existência prévia na realidade, pelo que antes de, por exemplo, qualquer observação, terão que ocorrer fases de (1)

identificação de um problema e/ou necessidade, (2) sugestão de uma solução e (3) **desenvolvimento** dessa (nova) solução.

O STIGH corresponde a uma nova tecnologia.

O fenómeno aqui em investigação não está categorizado *a priori*: por exemplo, o «objecto» *hiperligação stigmérgica* e os seus comportamentos de *reforço* e de *enfraquecimento* são novos.

O paradigma que se adequa a este processo de estudo é o **paradigma de Design** (Vaishnavi & Kuechler 2004) ou *Design Research* – que, em termos de enquadramento filosófico, não pode ser considerado tipicamente quantitativo. A diferença epistemológica entre as perspectivas em discussão, está que em *Design Research* (DR) o conhecimento constrói-se e *depois* observa-se, objectivamente.

Sendo uma mudança de estado uma alteração em algum valor caracterizador de um objecto, então, ontologicamente, cada componente do sistema desenvolvido, pode ser visto como podendo ter múltiplos estados, sempre correspondentes à mesma entidade.

A metodologia na perspectiva DR implica assim actividades de desenvolvimento e de mensuração do impacto, daquilo que é desenvolvido, no «mundo».

A Tabela 3 sintetiza os paradigmas quantitativo e DR. Está **destacado** o paradigma com o qual a investigação mais se identifica.

	<b>Paradigma Quantitativo</b>	<b>Paradigma de Design (DR)</b>
Ontologia	Realidade objectiva.	Realidade objectiva. Múltiplos estados.
Epistemologia	Investigador distante e independente do que é investigado.	Investigador desenvolve. Investigador distancia-se e mede o impacto do que criou.
Axiologia	Valores à margem. Não se assumem tendências. São de valor axiomas e formas de generalização.	São de valor: a criação, a melhoria do que foi criado, o entendimento do impacto da criação.
Retórica	Linguagem formal.	Linguagem formal.
Metodologia	Observação, quantificação, análise estatística.	Desenvolvimento. Mensuração do impacto.

Tabela 3 – Paradigmas quantitativo e DR

(Fallman 2003) argumenta que a generalidade da investigação na área científica de HCI (*Human Computer Interaction*), está centrada no desenvolvimento de *novos* sistemas de interação e comunicação.

Pela abordagem aqui proposta, a emergência de hiperligações que correspondam às preferências dos utilizadores de um website, não é algo que possa ser experimentado **antes** da concepção do STIGH, inquirindo, pelo que os métodos do **inquérito** e da **experiência** (sem sistema), não têm sentido inicial.

Para além disto, para se justificar a expressão «emergência de preferências», um hipotético inquérito (de descoberta de preferências) deveria ser implícito ou «invisível», o que parece esotérico. Já os novos *hyperlinks* propostos são basicamente indistinguíveis, para o utilizador, dos *hyperlinks* convencionais, para além de que operam «silenciosamente».

Quanto a experimentarem-se diferentes configurações de sistema (múltiplos estados de execução), essa possibilidade é um facto, pelo desenvolvimento do STIGH, que pode ser visto como um mecanismo para inquéritos invisíveis aos seus utilizadores. Fica assim sublinhada a importância primária do desenvolvimento do STIGH, **antes** de outros métodos.

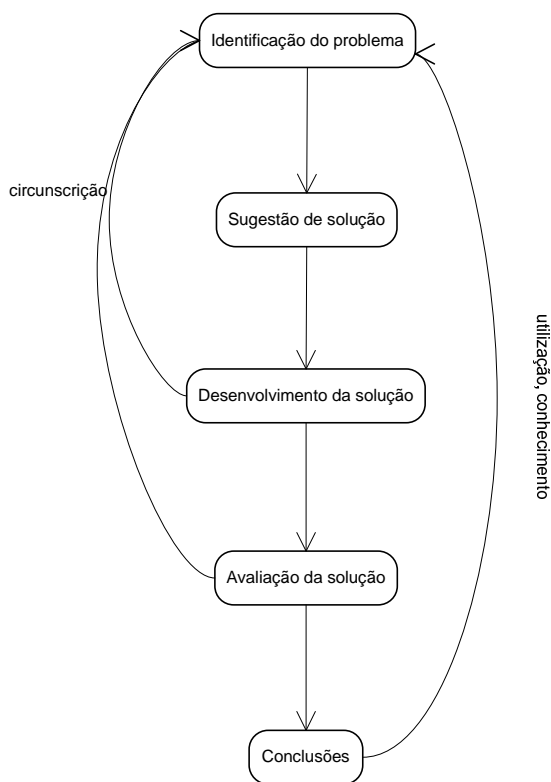


Figura 1 – Metodologia no paradigma DR. Adaptado de (Vaishnavi & Kuechler 2004)

No paradigma DR, os passos do processo de estudo, conforme ilustrados, consistem na (1) identificação do problema, (2) sugestão de solução, (3) desenvolvimento, (4) avaliação e (5) conclusões.

Esta sequência não é necessariamente linear: nas fases de desenvolvimento e de avaliação, podem identificar-se problemas circunscritos a estados específicos do sistema, que requerem reconhecimento e solução; é por isso que a figura tem transições de «circunscrição».

Mesmo após uma maturação que permita conclusões, o conhecimento que nasce da utilização do sistema desenhado pode justificar nova iteração: na perspectiva de *Design Research*, a axiologia valoriza a criação e o aperfeiçoamento.

Para cada passo esperam-se os seguintes resultados, descritos na Tabela 4.

<b>Passo</b>	<b>Resultado</b>
Identificação do problema	<p>Identificação de um problema/necessidade, que justifique a <b>criação</b> de um sistema, ou então a identificação de situações que, num sistema criado, justifiquem <b>aperfeiçoamento</b>.</p> <p>Neste caso: como capturar, transparentemente, as preferências de navegação dos utilizadores de um website, seguindo uma aproximação stigmérgica?</p>
Sugestão de solução	<p>Modelação de uma solução <b>formal</b>, capaz de responder ao problema de criação ou de aperfeiçoamento.</p> <p>Neste caso: o sistema STIGH.</p>
Desenvolvimento	<p>«<b>Materialização</b>» da solução formal, numa forma que permita a sua aplicação <b>objectiva</b> ao problema, com <b>medidas</b> de utilização.</p> <p>Neste caso um protótipo: a implementação do STIGH, na forma de um produto de software, efectivamente utilizável em websites. O sistema mantém diversos registos de actividade que correspondem a medidas de utilização.</p>
Avaliação	<p><b>Confronto</b> da solução criada ou aperfeiçoada, com o problema inicial, com a solução sugerida e com os resultados medidos na utilização.</p> <p>Neste caso: observação do que acontece às hiperligações stigmérgicas, num website com STIGH, ao longo do tempo.</p>
Conclusões	Retiradas da avaliação.

Tabela 4 – Passos da investigação e seus resultados

## 4. Modelação

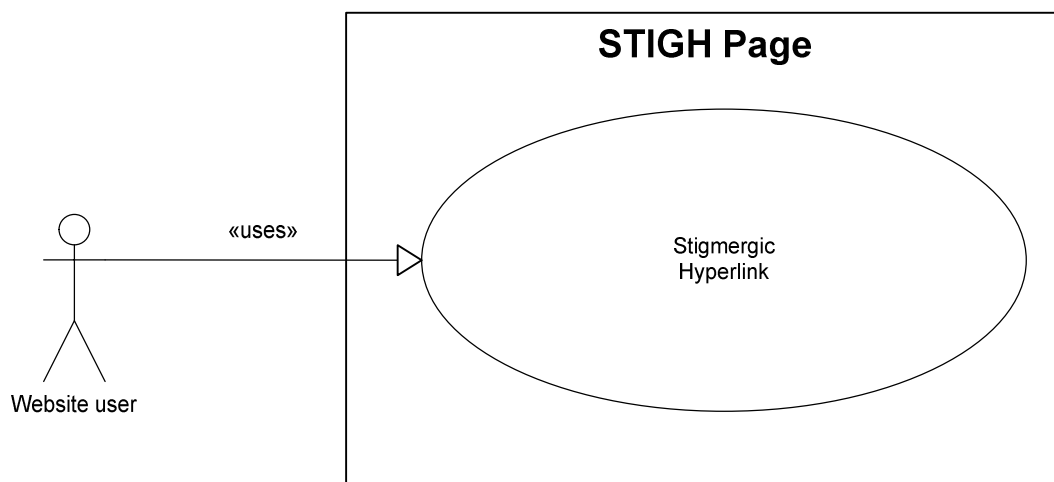


Figura 2 – Diagrama de casos de utilização

Quanto a casos de utilização, o sistema STIGH prevê apenas a selecção de hiperligações, na página. É só isso que os seus utilizadores podem fazer.

Seguindo uma aproximação stigmérstica e orientada a objectos, só é necessária uma classe, para as hiperligações e para o sistema: a essa classe foi dado o nome **cStigh**.

Numa primeiríssima modelação, para além de **cStigh**, chegou a estar pensada uma segunda classe «omnisciente» que, entre outros dados, manteria uma lista das hiperligações na página e saberia responder a pedidos de contagem como «quantos clicks já aconteceram?». Esta segunda classe poderia ter facilitado o desenvolvimento, mas teria comprometido a genuinidade da stigmergia, porque funcionaria como uma entidade central, sabedora e controladora: por exemplo, seria «ela» a controlar a inserção e a remoção de ligações na (sua) lista.

Foram desenvolvidas algumas outras classes de suporte. Os nomes dados são **cStighDemo**, **ClassAmUtil** e **ClassAmRandom**.

**cStighDemo** corresponde a um novo tipo de objectos, destinado a simular utilizações das ligações stigmérsticas numa página, permitindo testes aleatórios com qualquer número de clicks. A sua construção em nada compromete a classe principal **cStigh**.

As outras classes são depósitos de ferramentas utilitárias. **ClassAmUtil** providencia funções relacionadas com o consumo de recursos externos a uma página, como páginas noutros websites.

**ClassAmRandom** fornece o necessário para a realização de sorteios, como o sorteio em roleta proporcional.

Para compreensão do diagrama de classes, adiantam-se detalhes da programação do sistema, discutidos na secção 9:

- **cStigh** e **cStighDemo** foram concretizadas na plataforma Microsoft .NET;
- a classe **cStigh** é uma classe herdeira de **LinkButton**, que assume compromissos com as interfaces **IComponent**, **IDisposable**, **IParserAccessor** e **IDataBindingsAccessor**;
- a classe **cStighDemo** é uma classe herdeira de **Button**, que assume compromissos com as interfaces **IComponent**, **IDisposable**, **IParserAccessor** e **IDataBindingsAccessor**.

Uma interface é como que um contracto. Uma classe que assume uma interface, assume as responsabilidades impostas pelo contracto. As interfaces assumidas pelas duas classes mencionadas são uma imposição formal do ambiente tecnológico utilizado e não são discutidas neste trabalho.

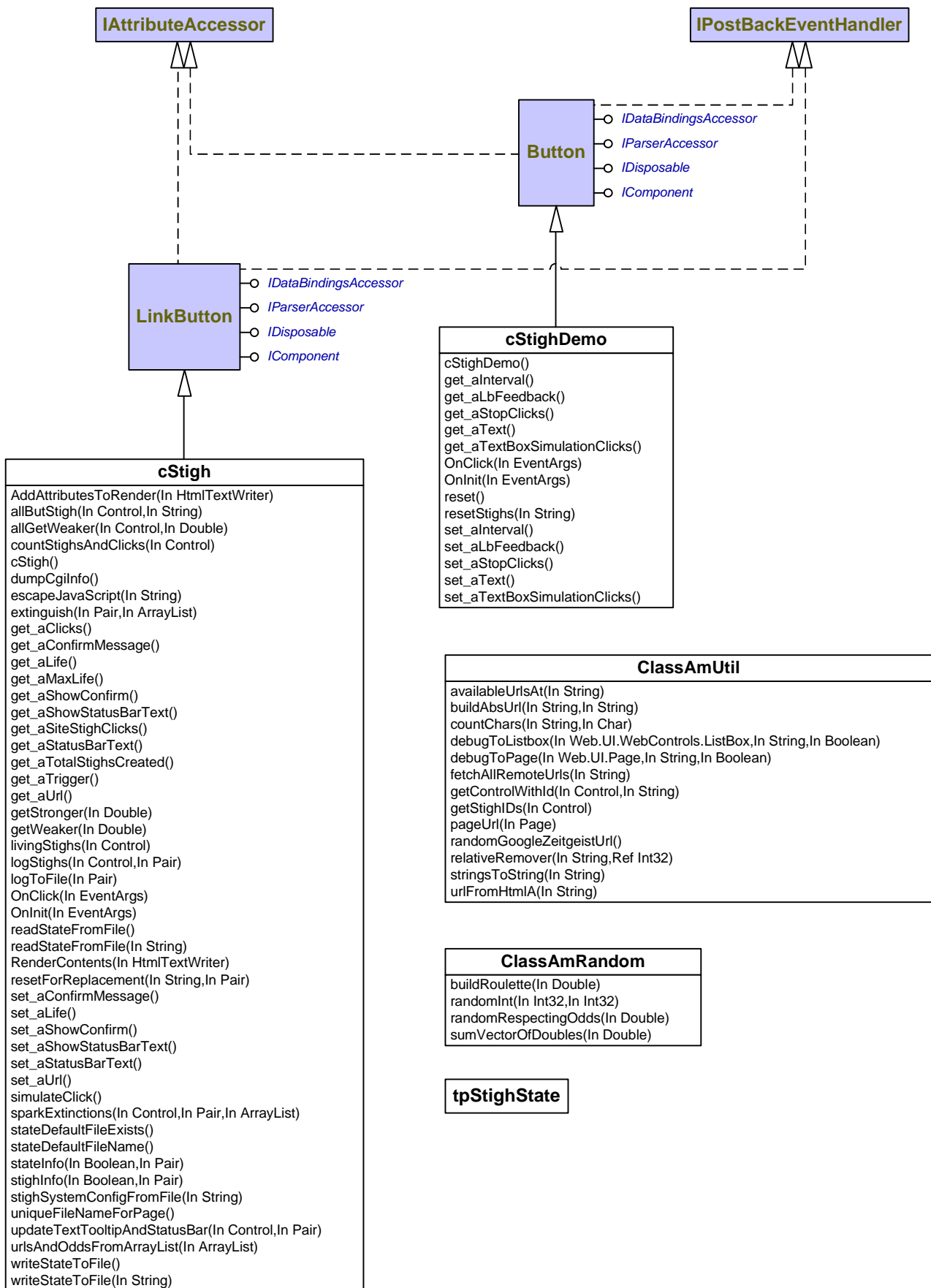


Figura 3 – Diagrama de classes

O essencial a reter sobre cada instância da classe **cStigh** – portanto sobre cada hiperligação stigmérgica – é:

- tem um estado interno, correspondente à sua vitalidade, ao seu URL de destino e ao seu contador de clicks;
- sabe responder aos métodos públicos
  - **getStronger**, pelo qual ganha vitalidade;
  - **getWeaker**, pelo qual perde vitalidade;
  - **readStateFromFile**, pelo qual recupera a sua informação de estado;
  - **writeStateToFile**, pelo qual preserva a sua informação de estado;
  - **stateDefaultFileExists**, pelo qual responde se existe o seu ficheiro de estado;
  - **stateDefaultFileName**, pelo qual responde o nome do seu ficheiro de estado;
  - **uniqueFileNameForPage**, pelo qual identifica a sua página;
  - **simulateClick**, que equivale a um click.

A discussão de todos os métodos (não apenas os públicos), propriedades e membros de dados, é feita na secção 9.

Em anexo também se disponibilizam diagramas de fluxo (*flow charts*) e diagramas de sequência (*sequence charts*), para todos os métodos, de todas as classes. O anexo não é fornecido em papel, porque os conteúdos em causa não se adaptam a esse suporte. Para além da versão de demonstração do STIGH – que só é possível online –, muitas das imagens dos diagramas, têm uma altura e/ou comprimento superior a 3000 pixéis, o que obrigaria a várias páginas para imprimi-las ou então a uma redução tal, que ficariam ilegíveis.

## 5. Hipertexto e hiperligações

### 5.1. Origens e evolução

Este é um trabalho na área de hipertexto, propondo uma nova forma de hiperligação, capaz de adaptar-se, transparentemente, à comunidade de utilizadores de um website. Escrito isto, é interessante notar a evolução do significado de «hipertexto» e a origem do termo «hiperligação».

O conceito de hiperligação tem sido central nesta área de estudos mas, curiosamente, *não* surge no primeiro trabalho, de 1965, de Theodor Nelson – *A File Structure for the Complex, the Changing, and the Indeterminate* –, que estreia as palavras *hypertext*, *hyperfilm* e *hypermedia* (Nelson 1965). O artigo em causa apresenta uma estrutura para ficheiros, denominada ELF (*Evolutionary List File*), orientada à representação de documentos que não poderiam ser convenientemente representados em papel, pela forma complexa como se relacionariam os seus componentes de texto e/ou pictóricos.

Em 1970, no artigo *No More Teachers' Dirty Looks* (Nelson 1970), sobre educação assistida por computador, Theodor Nelson foca-se nos neologismos de 1965 e identifica alguns tipos de *hypermedia*: (1) *discrete hypertexts*, (2) *hypergrams*, (3) *stretchtexts*, (4) *hypermaps*, e (5) *queriable illustrations*.

*Discrete hypertexts* correspondem a formas de escrita que se ramificam ou desempenham, a pedido do leitor/utilizador. O artigo escreve que “*hypertexts consist of separate pieces of text connected by links*”; ou seja, que hipertextos consistem em porções de texto, conexas por *links*. Esta é tida como a primeira utilização da palavra *link*, no contexto de *hypertext*.

*Hypergrams* correspondem a material pictórico que se ramifica ou desempenha, a pedido do utilizador.

*Stretchtexts* são uma forma de hipertexto, sem *links*, cuja visualização pode ser controlada para cima e para baixo, num monitor (acção que hoje pode ser desempenhada pelas *scrollbars*); e também pode ser controlada, quanto ao tamanho das letras utilizadas na escrita.

*Hypermaps* são uma forma de interface receptiva a aproximações (*zoom-ins*) e afastamentos visuais (*zoom-outs*).

*Queriable illustrations* (imagens receptivas a questões) são um caso particular de *hypermaps*.

Quando se escreve que certas formas de *hypermedia* se ramificam ou desempenham, a pedido, o que isso quer dizer é que os conteúdos respectivos podem perder a linearidade em determinados pontos (*links*, no caso de hipertexto), e que permitem acções como *scroll* ou *zoom*, ou outras.

Com base nestes primeiros trabalhos de Theodor Nelson, Noah Wardrip-Fruin chega a uma definição: «hipertexto é um termo cunhado por Ted Nelson, para formas de hipermédia, que operam sobre texto» (Wardrip-Fruin 2004). O exemplo mais comum é a WWW, que é um sistema de hipertexto discreto, baseado em *links*. Na prática, passou a chamar-se a essas ligações *hyperlinks*, ou hiperligações.

As implicações da definição, de acordo com Wardrip-Fruin, são que, ao longo dos anos, a informação ligada foi assumindo formas muito para lá de «porções» textuais e/ou pictóricas. Também a investigação em hipertexto trabalhou, essencialmente, em tecnologias e ferramentas (e não em conteúdos). Por exemplo, este trabalho é sobre uma nova tecnologia de hiperligações.

Feita esta viagem ao conceito original de hipertexto, fica claro que o significado do termo se dilatou, significativamente, na sua abrangência, tanto na comunidade de autores de conteúdos, como na comunidade de autores de tecnologias e ferramentas.

## 5.2. Hipertexto doméstico e hipertexto selvagem

Alguns da investigação mais interessante em hipertexto, tem sido a proposta de sistemas com particularidades que vão ao encontro de necessidades específicas, não satisfeitas pelas soluções ubíquas, como a hiperligação em HTML.

Por exemplo, Jim Rosenberg, propõe o *Frame Stack Project*, que tem como objectivo primeiro, o suporte de «estruturas selvagens», para objectos de hipertexto (Rosenberg 2000). A inspiração de Rosenberg é o bloco de notas de um poeta: um suporte de informação raras vezes fechado, em que, com frequência, os conteúdos são anotados, sem que esteja claro o seu destino/propósito, permanecendo não relacionados entre si.

A adjectivação de «selvagem», para algumas situações em hipertexto tem vindo a crescer, com significados diferentes, principalmente desde a popularização de websites sociais, com sistemas de classificação completamente livres (Cameron Marlow, Danah Boyd, Marc Davis 2006).

No caso do *Frame Stack Project*, a natureza «selvagem» está na possibilidade dos objectos do sistema poderem fugir da jaula (doméstica), que é o seu ambiente de desenvolvimento, e poderem existir «livremente» no *desktop* do utilizador, sem que lhes esteja atribuído um destino. Ou seja, o utilizador tem a possibilidade de adiar a criação da estrutura, mantendo a nota «solta». Poderia discutir-se se a estrutura não existe desde logo, ainda que incompleta, mas o objectivo desta referência é a classificação de «selvagem» (*feral*, no original).

De acordo com Jill Walker, hipertexto selvagem (*feral hypertext*) corresponde às situações em que os conteúdos começam a deixar de estar sob controlo de alguma entidade central e evoluem em sentidos potencialmente imprevisíveis: «os melhores exemplos de *feral hypertext* são os projectos colaborativos de grande escala, nos quais emergem padrões e significados, sem que existam autores ou editores, controlando as ligações» (Walker 2005).

Alguns casos particulares e populares, muito evidentes, são os websites <http://del.icio.us> e <http://flickr.com>. No primeiro caso, os utilizadores contribuem com URLs de websites que apreciam e classificam-nos de forma completamente livre; no segundo caso, passa-se exactamente a mesma coisa, mas relativamente à submissão de imagens. Isto significa que vigora não uma taxonomia centralizada, mas uma *folksonomia* (*folksonomy*, no original – um termo atribuído a Thomas Vander Wal) descentralizada pela comunidade: um sistema de classificação colaborativo e aberto.

Os textos, ou marcas de classificação, ou *tags*, que vão emergir da comunidade, são totalmente imprevisíveis. Por exemplo, é muito habitual que novas tecnologias com alguma expressão gráfica, convidem os seus utilizadores a partilharem, via [flickr.com](http://flickr.com), as imagens correspondentes.

Um caso é o serviço «*websites as graphs*» (Sala 2006), que mostra websites enquanto grafos. Os utilizadores são convidados a enviarem o grafo correspondente ao seu website, para [flickr.com](http://flickr.com), com a marca/tag **websitesasgraphs**. No caso de uma taxonomia, teriam os autores do sistema de classificação pensado nessa possibilidade? Em caso negativo, seriam ágeis a actualizar a taxonomia, para que a incluísse? Dificilmente. Uma vantagem dos sistemas colaborativos já referidos é a sua agilidade. Ao serem totalmente abertos à colaboração, crescem em sentidos «selvagens».

O sistema de hiperligações stigmérgicas (STIGH) também é colaborativo, no sentido de que a dinâmica das hiperligações que o compõem, é determinada pela comunidade de utilizadores da página. É um facto que existe uma mecânica subjacente ao sistema, que por exemplo determina a intensidade e a regularidade das recompensas e das penalizações das ligações, mas essa mecânica não faz mais do que sustentar as interacções que venham a acontecer, da mesma forma que o sistema de contribuições de *tags*, da generalidade dos sites sociais, admite que o utilizador possa etiquetar/marcar os seus próprios conteúdos, mas não possa eliminar marcas alheias.

Sendo um sistema colaborativo, que cria condições para a emergência das preferências da comunidade dos utilizadores da página, é também um sistema «selvagem», que fica em liberdade, a partir do momento que a página fique online.

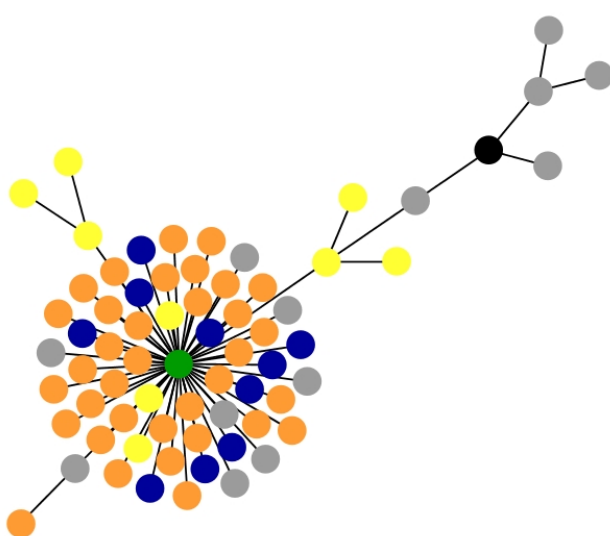


Figura 4 – *websitesasgraphs* para a página de demonstração do STIGH, em <http://stigh.org>

À esquerda, o grafo de **websitesasgraphs**, para a página de demonstração do STIGH, disponível em <http://stigh.org>.

As cores representam marcas no HTML.

**azul:** links (A tag)

**verde:** DIV tag

**amarelo:** FORM, INPUT, TEXTAREA, SELECT, OPTION tags

**laranja:** BR, P, BLOCKQUOTE tags

**cinzento:** outras tags

**preto:** HTML tag (nó raiz).

### 5.3. Outros sistemas de hiperligações

Um esforço concertado no sentido de *hyperlinks* mais capazes do que as actuais ligações unidireccionais, é a **XML Linking Language** (W3C 2001). Esta linguagem permite a declaração de elementos (**XLinks**) para a conexão de recursos, que podem comportar-se como as ligações actuais ou outras mais sofisticadas. Algumas das diferenças previstas são (1) a possibilidade de associação de hiperligações a meta-dados, e (2) a verificação da validade da ligação.

As hiperligações stigmérgicas propostas nesta investigação podem muito facilmente ser modificadas para se recusarem a ligar a recursos que tenham desaparecido (e então ligarem a outro destino, ou sinalizarem a invalidade de alguma forma). Mas essa funcionalidade e as opções de desenho relacionadas, foram remetidas para trabalho futuro, porque o foco presente é ilustrar a funcionalidade do sistema elementar, do qual pode esperar-se que, ao longo do tempo, as ligações «falsas» sejam *naturalmente* eliminadas, por não capturarem as preferências de navegação.

A valorização, sem reinvenção, das hiperligações actuais, é procurada em trabalhos centrados em *linkbases* (bases de dados de *hyperlinks*), utilizadas como repositórios de ligações. Sobre as *linkbases* podem funcionar níveis lógicos, orientados a diferentes serviços, como a partilha de *hyperlinks* entre utilizadores e grupos de utilizadores, com interesses semelhantes. Neste último caso, a *linkbase* é explorada como um sistema de recomendações (*recommender system*) (Samhaa R. El-Beltagy, David De Roure, Leslie Carr 2001).

Sistemas abertos de hipertexto, deverão permitir uma utilização de hiperligações «livre», entre programas (L. A. Carr1, W. Hall, G. Hill 1998). Isto significa que qualquer aplicação no sistema deverá poder aceder a um serviço de hiperligações, que trate de operações de manutenção de *hyperlinks*, como procura, validação e actualização de recursos.

O *Open Link Service* (OLS) de Carr, De Roure, Hall e Hill, propõe-se neste sentido (L. A. Carr1, W. Hall, G. Hill 1998). De forma a consegui-lo, as ligações são geridas separadamente dos documentos – numa *linkbase* – de maneira a que possam ser aplicadas, em teoria, sobre qualquer formato. A expressão *linkbase* surgiu a propósito do sistema **Microcosm**, destes mesmos autores.

Na prática, a implementação feita (DLS – *Distributed Link Service*) produz documentos nos formatos HTML, RTF e PDF. A injeção das ligações nos documentos é um processo chamado de «compilação». Diferentes parâmetros de compilação de documentos, poderão injectar diferentes hiperligações, apropriadas a audiências distintas.

Entre as facilidades que o DLS disponibiliza, estão ligações genéricas (*generic links*) – tidas como apropriadas à generalidade das audiências –, e funções de busca de informação, sobre a *linkbase*. A busca pode ser feita recorrendo a uma interface cliente, via browser, pois o serviço está arquitectado como uma aplicação CGI (*Common Gateway Interface*).

Os *generic links* são especialmente úteis para nomes de pessoas, lugares e conceitos «estabelecidos», pois basta inseri-los na *linkbase* e, de aí em diante, estarão automaticamente disponíveis nos sítios em que as expressões associadas aparecerem, nos documentos a compilar.

Um utilizador pode também pesquisar, pela interface via browser, por *hyperlinks*, utilizando palavras-chave.

Com um objectivo muito mais localizado, está disponível o sistema **Insipid** (Reeves 2005). O **Insipid** é um programa, escrito em **Perl** e disponibilizado gratuitamente em *open-source*, que funciona como uma *linkbase* para utilizadores individuais; isto é, todos os links serão «propriedade» de um só utilizador. A base de dados manipulada tem que ser servida via **MySQL** ou **PostgreSQL**. O **Insipid** funciona no contexto de um servidor http, como **Apache** ou **IIS**.

Uma característica interessante do **Insipid** é que as hiperligações devem ser classificadas com marcas/*tags*, como em <http://del.icio.us>.

Uma vez que o próprio [del.icio.us](http://del.icio.us) pode ser visto como uma grande *linkbase*, **Insipid** apresenta-se como um *delicious clone*.

**Insipid** não permite a compilação de *links*, muito menos *generic links*, em documentos, mas permite uma função de procura, pelo texto da ligação. Também é possível procurar por marcas, mas não textualmente: só recorrendo à lista de *tags*, sempre visível durante a navegação.

A «invulgaridade» de **Insipid** é que está concebido para um só utilizador. Por isso, talvez seja abusivo falar de *folksonomia* neste sistema (*individualnomy / individualnomia?*).

Apesar da diversidade de trabalhos, académicos e outros, em redor do conceito de hiperligação, este estudo não conseguiu identificar nenhum outro, em que se proponha um objecto *hyperlink*, de inspiração stigmérgica. Aliás, não foram identificados trabalhos centrados na reformulação de objectos de hipermédia, de forma a resolver, por emergência estritamente centrada no objecto, alguma questão da área.

## 6. Utilização (ilustrada) do STIGH

Segue-se uma sequência, ilustrada, de utilizações de uma das páginas de demonstração do STIGH. O propósito é explicar, com a ajuda da informação que as próprias hiperligações stigmérgicas vão produzindo, o que está a passar-se.

No canto superior direito de cada imagem, está um número a verde, que indica quantos clicks já ocorreram, na página.

A sequência ilustrada que se segue é retirada da própria aplicação desenvolvida, que está em regime experimental evolutivo na Web, em <http://stigh.org>.

## 6.1. Início



Figura 5 – Exemplo ilustrado de utilização do STIGH - inicialização

A primeira imagem mostra um sistema com 10 **cStighs**: 10 hiperligações stigméricas. Estes objectos têm um identificador (**ID**) unívoco que, no sistema de demonstração, é **CS $n$**  para o link de ordem  $n$ : CS1 para o primeiro (topo da página), CS2 para o segundo, ... , CS10 para o décimo (último link).

Não existem hiperligações convencionais na página.

Para efeitos de demonstração, este sistema acabou de iniciar-se. Ao iniciar-se, o sistema lê três valores relevantes, do ficheiro **stigh.cfg**, presente na raiz do website que serve a página. O primeiro valor (no caso, a barra vertical | ) só tem efeito estético e estabelece a frase que separa certas escritas em *tooltip* e em *statusbar*; o segundo valor (no caso, **2**) estabelece o valor interno **iTrigger** que dita a cadência do envelhecimento natural da página – a cada dois clicks, todos os *links* ficarão com menor vitalidade –; e o terceiro valor (no caso, **100**) estabelece a vida máxima de qualquer **cStigh**.

Cada **cStigh** informa do seu estado por dois mecanismos, ambos desencadeados quando se estaciona o ponteiro do rato em cima da representação da ligação: (1) escrevendo o estado na zona de *statusbar* do browser (canto inferior esquerdo da imagem), e (2) mostrando o seu estado, em *tooltip* (zonas de fundo amarelo suave, na imagem).

A imagem foi trabalhada para mostrar as *tooltips* de todos os **cStigh** – mas, em real utilização, só é visível uma *tooltip* a cada instante. Na imagem, a informação em *statusbar*, como está cristalizada, diz respeito apenas à hiperligação que, eventualmente, tenha estado a sofrer um evento de *mouseover*, no momento de captura (no momento de *PrintScreen*).

Imagine-se que o visitante da página vai escolher / seleccionar / fazer click na ligação <http://formula1.com> . O visitante será remetido para o website escolhido, caso confirme essa escolha, na caixa de diálogo. A caixa de diálogo é uma opção do autor da página: um **cStigh** pode *não* exibir esse comportamento de pedido de confirmação da navegação.

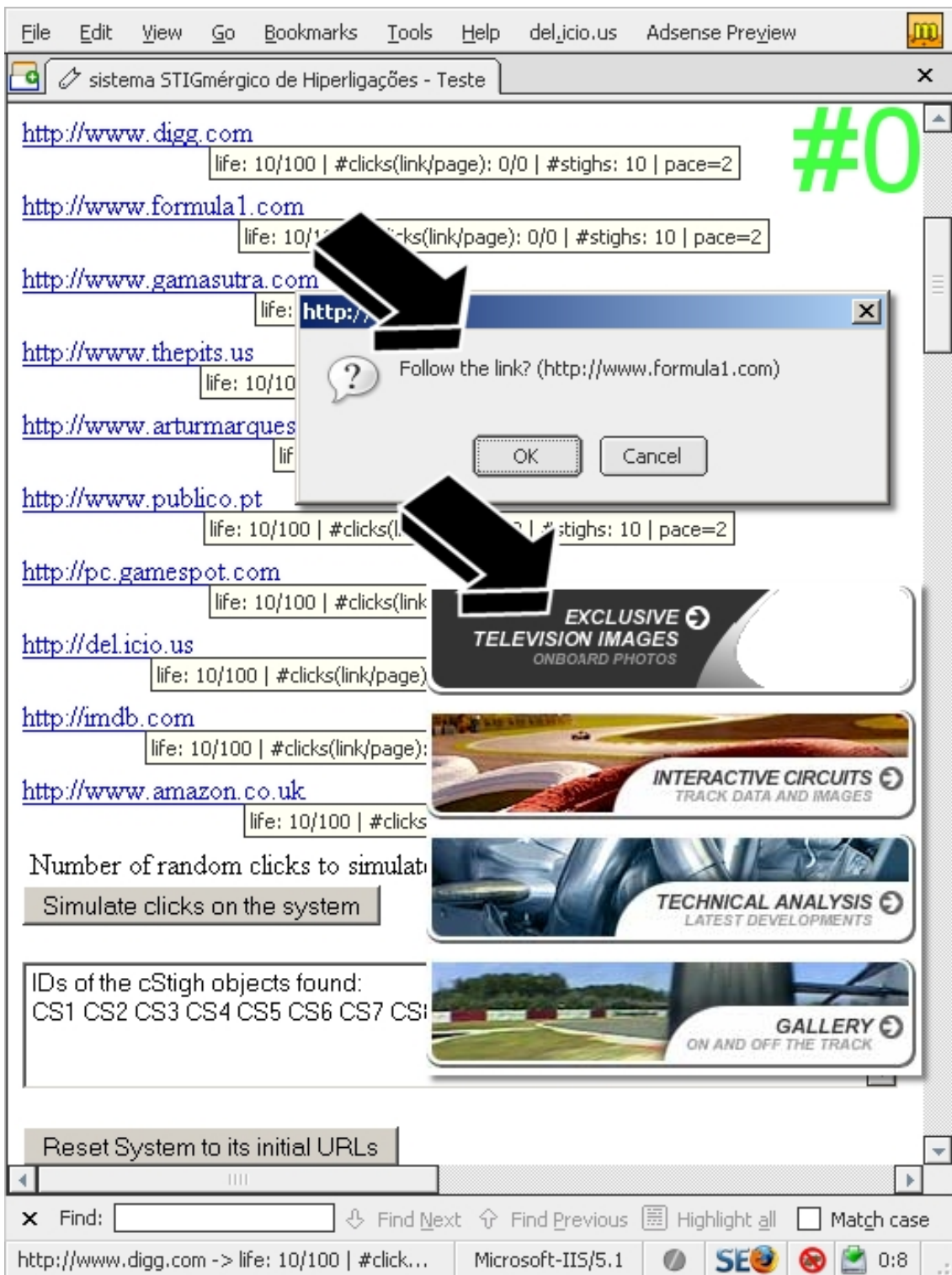


Figura 6 – Exemplo ilustrado de utilização do STIGH - sequência do primeiro click

## 6.2. Após um primeiro click

Internamente, o sistema acusa a selecção feita, recompensando o **cStigh** escolhido com  $2 * iTrigger = 2 * 2 = 4$  pontos de vitalidade. O utilizador não vê essa recompensa, porque a janela muda para o website seleccionado. Todavia, se o utilizador voltar atrás e pedir a actualização da página (*reload/refresh*), essa mudança no estado interno da ligação torna-se visível, conforme a imagem seguinte documenta. Assume-se que nenhum outro utilizador utilizou a página, no entretanto.



Figura 7 – Exemplo ilustrado de utilização do STIGH - um click ocorrido

O que agora pode ler-se é a vitalidade de 14 (em 100), para o objecto com **ID CS2**, que é o **cStigh** que liga a <http://www.formula1.com> . Nenhum dos outros **cStigh** perdeu vitalidade, porque só aconteceu um evento de click e o envelhecimento sistemático acontece apenas a cada duas selecções (a cada **iTrigger** selecções).

Também é possível ler-se quantas vezes cada ligação foi seleccionada e quantas selecções houve, ao todo, na página. Para todos os **links**, excepto para **CS2**, observa-se 0/1 (0 selecções no link, 1 selecção na página).

Assuma-se que, de seguida, o visitante selecciona <http://www.publico.pt> (**cStigh** com **ID CS6**). O que vai ocorrer tem consequências mais profundas do que o que se passou no primeiro click: para além da recompensa de 4 pontos de vitalidade para o **cStigh** eleito, uma vez que se está na segunda selecção (que é um múltiplo de **iTrigger=2**), acontece também uma quebra de vitalidade generalizada, no valor de metade da recompensa, ou seja no valor de  $(4/2 = 2)$ .



Figura 8 – Exemplo ilustrado de utilização do STIGH - sequência do segundo click

### 6.3. Após um segundo click

Assim, se o utilizador regressar e fizer *reload/refresh* da página, para testemunhar o efeito da sua acção, descobrirá o **cStigh CS6** com  $10+4-2=12$  pontos de vitalidade, e todas as outras hiperligações com menos 2 pontos do que antes, ou seja todas com 8 pontos, excepto <http://www.formula1.com>, que reduziu a sua «vida» para 12.

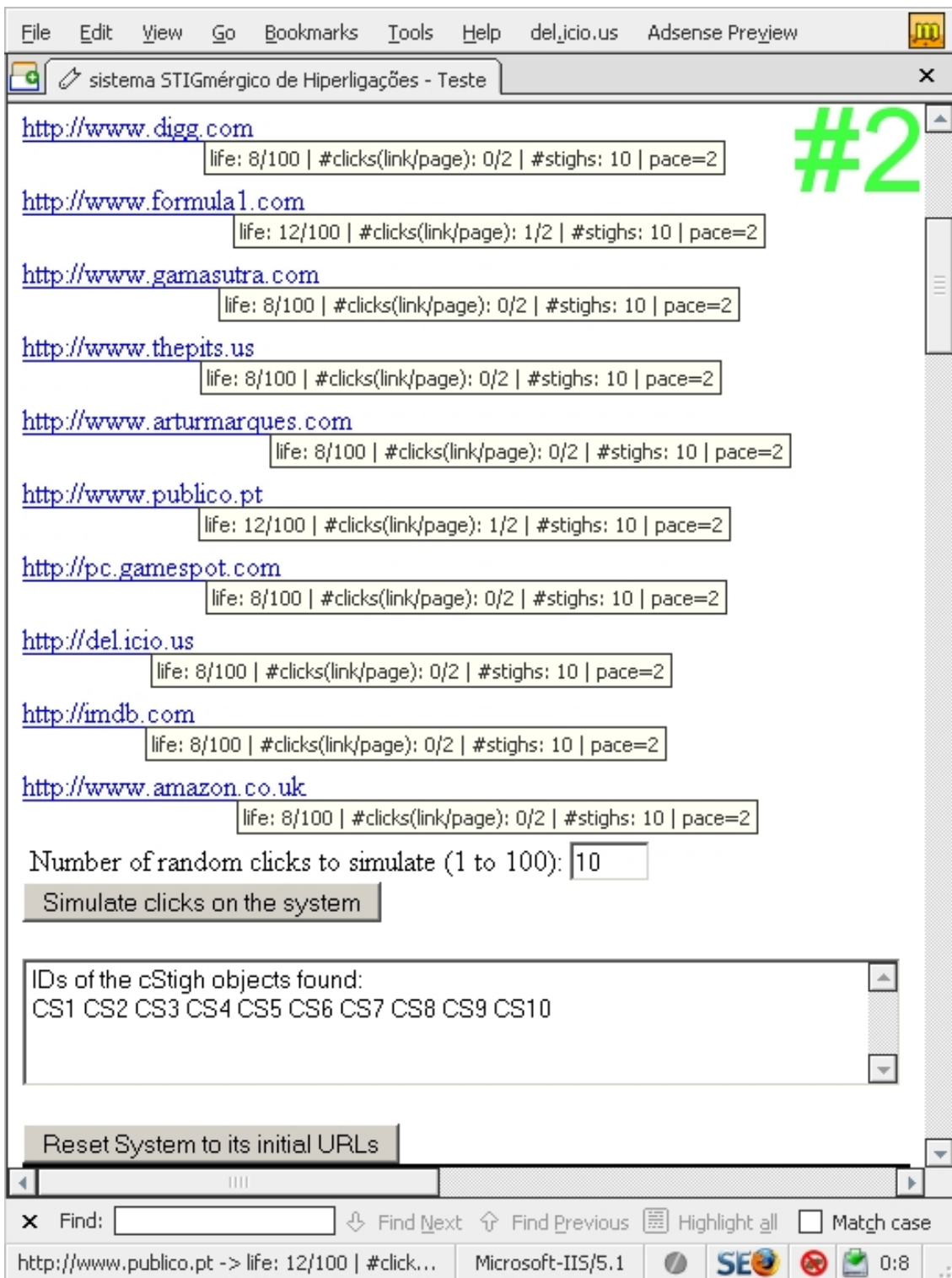


Figura 9 – Exemplo ilustrado de utilização do STIGH - dois clicks ocorridos

Note-se que a contagem de clicks na página acusa dois eventos e que estão correctos os contadores individuais de cada hiperligação.

#### **6.4. Após um terceiro click**

Para acelerar o exemplo no sentido de uma extinção, assumase que um outro visitante (ou o mesmo), visita (novamente) **CS2**. As contas são fáceis: a vitalidade de **CS2** passará de 12 para  $12+4=16$ ; e os seus clicks passarão para 2 em 3 (2 em **CS2**, 3 na página). As outras ligações não sofrem alterações de vitalidade, mas a sua contagem de selecções escreverá 0 em 3 (0 clicks, num total de 3 no sistema), excepto **CS6**, que escreverá 1 em 3.



Figura 10 – Exemplo ilustrado de utilização do STIGH - três clicks ocorridos

## 6.5. Após um quarto click

Na selecção seguinte, assuma-se que o link eleito é CS8 ( <http://del.icio.us> ). Por ser o quarto click, acontecerão reforços e penalizações. Será reforçado CS8 – a sua vitalidade passará para  $8+4=12$ . Mas todos serão penalizados em 2 pontos, pelo que CS8 ficará com a vida em  $12-2=10$ .

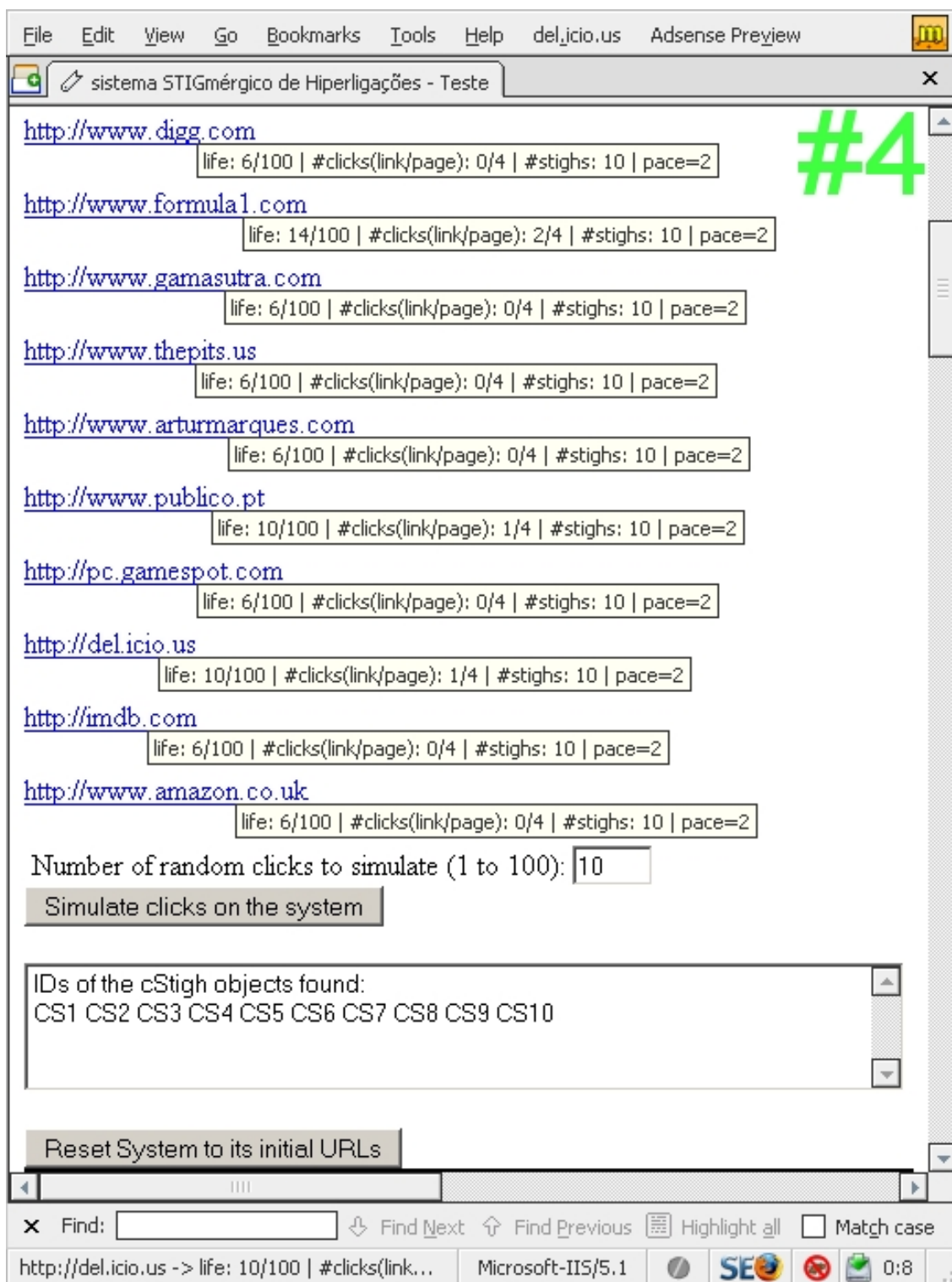


Figura 11 – Exemplo ilustrado de utilização do STIGH - quatro clicks ocorridos

É de notar o enfraquecimento gradual das ligações que ainda não capturaram as preferências do(s) visitante(s), neste exemplo. Mais algum tempo com esta tendência, e alguma(s) dela(s) serão extinta(s).

## 6.6. Quinto e sexto clicks

Admita-se que <http://www.gamasutra.com> (CS3) é a quinta selecção. As alterações são simples: a sua vitalidade salta para  $6+4=10$ . Nenhuma ligação é penalizada. O sistema acusa 5 clicks.

Seja <http://www.publico.pt> (CS6) a próxima preferência de algum visitante. As consequências são mais significativas: a sua vitalidade sobe para  $10+4-2=12$  pontos. Ao sexto click, ocorre mais uma degradação sistémica generalizada, no valor de 2 pontos de vitalidade. Assim as ligações mais fracas descem para  $6-2=4$  pontos. CS2 desce para  $12-2=10$  pontos. CS3 e CS8 descem para  $10-2=8$  pontos.



Figura 12 – Exemplo ilustrado de utilização do STIGH - seis clicks ocorridos

## 6.7. Sétimo e oitavo clicks

Rumo a evento(s) de extinção, admita-se que sucedem-se as selecções #7 em <http://pc.gamespot.com> (CS7) e #8, em <http://www.formula1.com> (CS2).

Em #7, CS7 é reforçado para  $4+4=8$  pontos. Nenhuma ligação perde vitalidade.

Em #8, CS2 é reforçado para  $12+4-2=14$  pontos. Todas as ligações perdem 2 pontos de vitalidade. Assim, CS1, CS4, CS5, CS9 e CS10, que permanecem sem conquistar uma única preferência, desvalorizam-se para  $4-2=2$ . CS3, CS7 e CS8 degradam-se para  $8-2=6$ . CS6 desce para  $12-2=10$ .

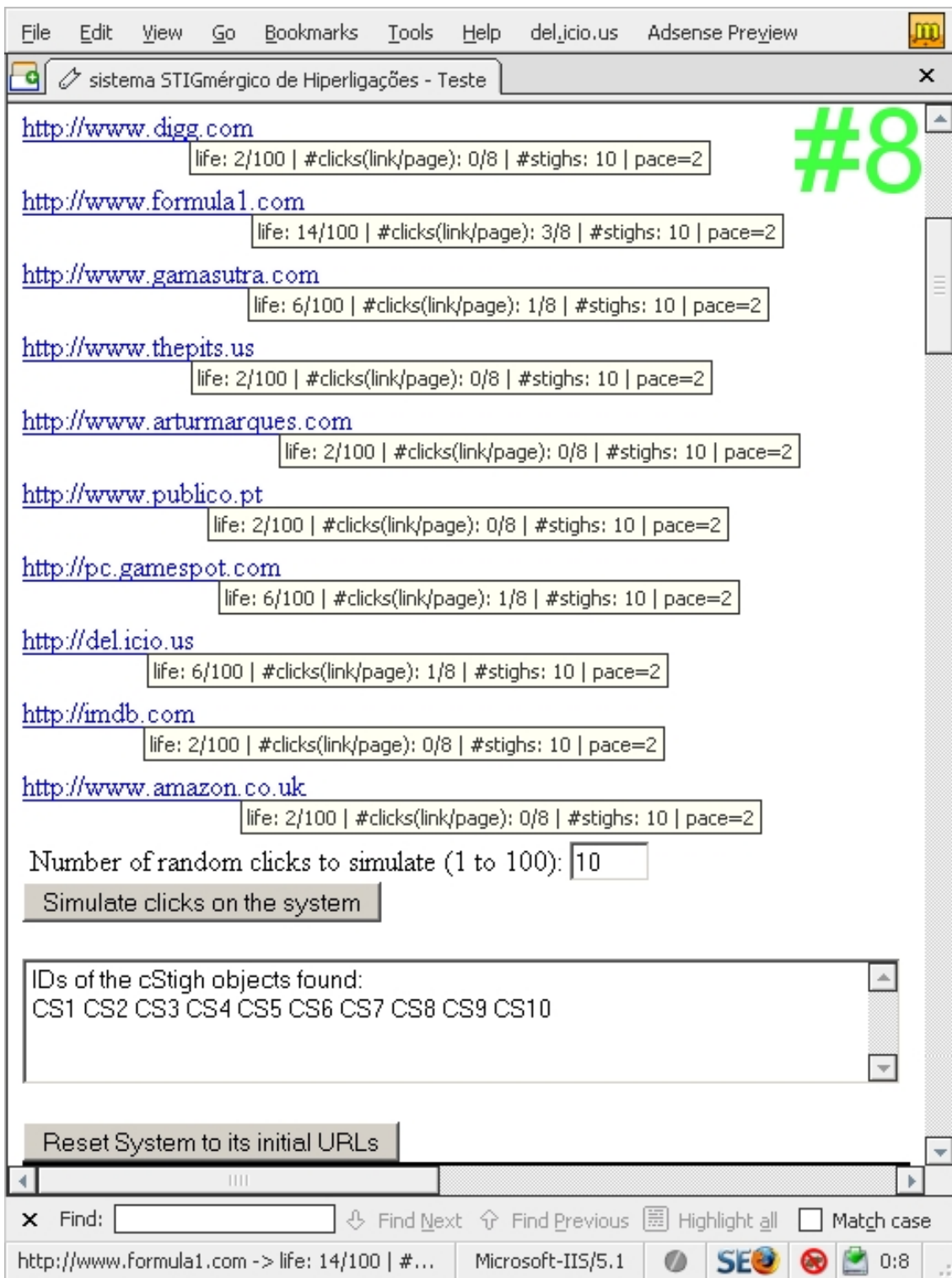


Figura 13 – Exemplo ilustrado de utilização do STIGH - oito clicks ocorridos

O exemplo está no limiar de desencadear extinções. Dentro de dois clicks, as hiperligações stigméricas que tiverem permanecido intocadas, ver-se-ão substituídas por recursos sorteados do universo de ligações dos **cStigh** sobreviventes. A ideia subjacente é que as ligações que foram conquistando preferências de navegação, deverão ligar-se a recursos potencialmente interessantes para a comunidade de utilizadores.

## 6.8. Ao nono click, antes de extinções

Admita-se que ao nono (#9) click, <http://arturmarques.com> (CS5) é a opção de algum visitante. O objecto é recompensado até aos  $2+4=6$  pontos de vida.

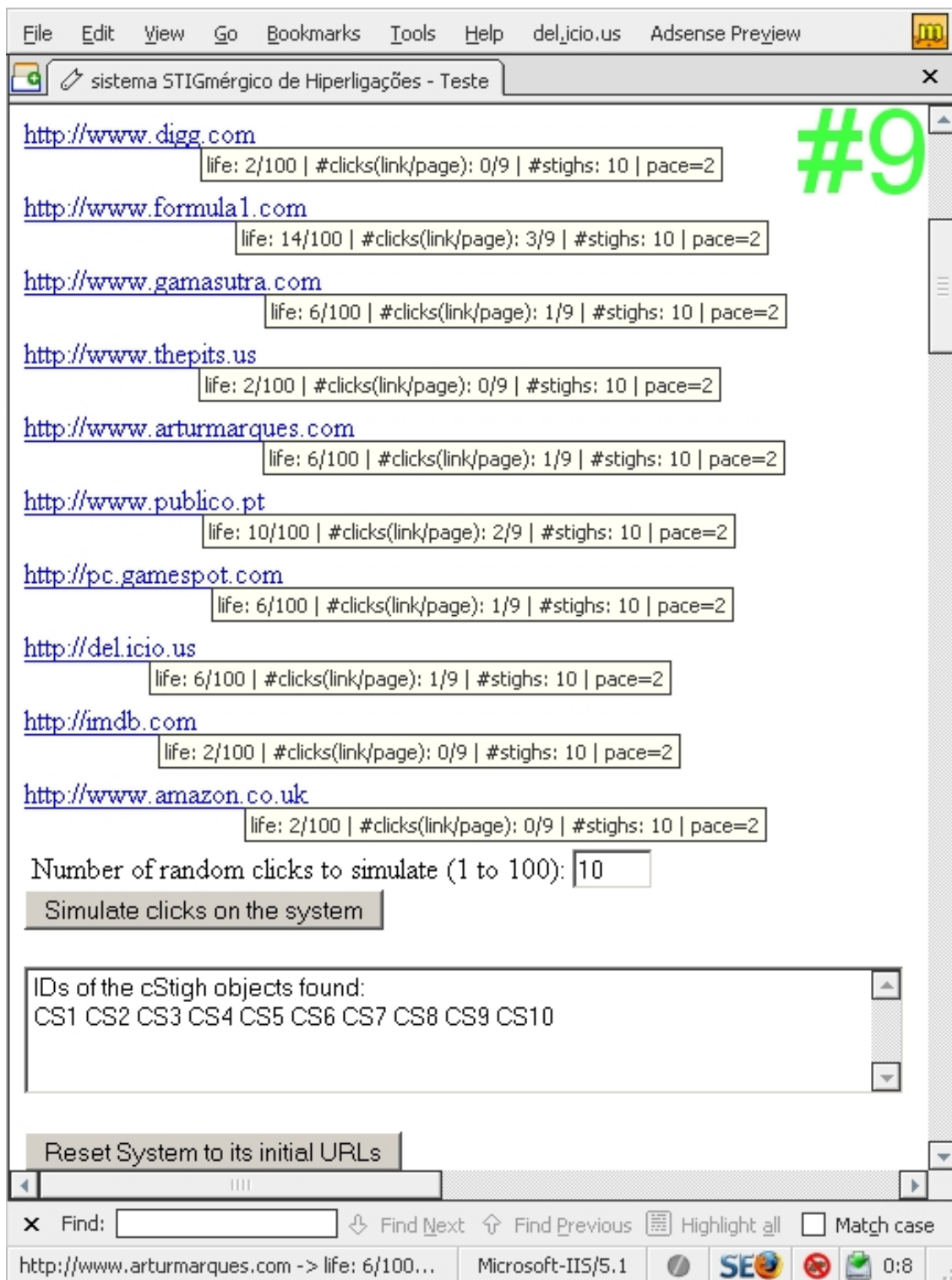


Figura 14 – Exemplo ilustrado de utilização do STIGH - imediatamente antes de extinções

## 6.9. Décimo click: a faísca para uma extinção tripla

Admita-se que ao décimo (#10) click, <http://www.digg.com> (**CS1**) é a opção de algum utilizador. Esse objecto é recompensado até aos  $2+4-2=4$  pontos, escapando à substituição, mesmo no limite. Quem *não* se furta à extinção, são **CS4**, **CS9** e **CS10**...

Ao acontecer o décimo click, **CS1** fica com uma vitalidade de  $2+2-2=2$  pontos. Todas as ligações perdem 2 pontos de «vida». Para **CS4**, **CS9** e **CS10**, isso significa  $2-2=0$  de vitalidade (vida nula). Estas ligações, perante a mensagem de **sparkExtinctions**, lançada por **CS1**, vão invocar o seu próprio comportamento de **extinguish**.

Ao extinguir-se, cada **cStigh** deve eleger um sucessor. O primeiro passo nesse sentido é sortear, em roleta proporcional, qual dos objectos sobreviventes deve fornecer o recurso substituto. A roleta diz-se proporcional, porque as suas «fatias» devem respeitar as preferências já capturadas, reflectidas na vitalidade de cada objecto.

## 6.10. Cálculo de probabilidades e roleta proporcional

No cenário do exemplo, o cálculo das probabilidades, respeitando a vida das ligações, é:

cStigh ID	dLife   Vida	Probabilidade	P
CS1	2	$2/\sum = 2/38 = 1/19$	1/19
CS2	12	$12/\sum = 12/38 = 6/19$	6/19
CS3	4	$4/\sum = 4/38 = 2/19$	2/19
CS4	0	0	0
CS5	4	$4/\sum = 4/38 = 2/19$	2/19
CS6	8	$8/\sum = 8/38 = 4/19$	4/19
CS7	4	$4/\sum = 4/38 = 2/19$	2/19
CS8	4	$4/\sum = 4/38 = 2/19$	2/19
CS9	0	0	0
CS10	0	0	0
Soma das vidas = $\sum =$	38	Soma das P =	1

Tabela 5 – Cálculo de probabilidades para a roleta proporcional

A esta distribuição de probabilidades, corresponde uma roleta proporcional. É notório que a maior fatia da roleta deve pertencer a **CS2**, por ter uma vitalidade consideravelmente superior à das restantes ligações.

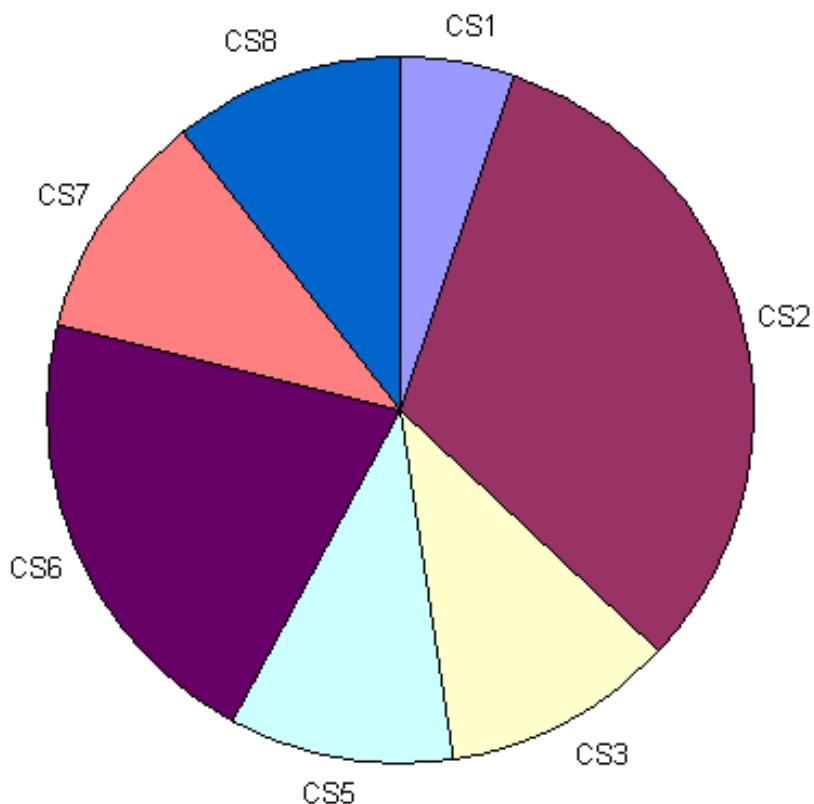


Figura 15 – Roleta proporcional a ser utilizada nas extinções, ao décimo click

Pelo facto de **CS2** ter direito a uma fatia de roleta maior, isso não garante que venha a ser a fonte da ligação substituta de algum dos **cStigh** a extinguir, embora probabilisticamente esteja bem posicionada nesse sentido.

No caso do exemplo ilustrado em curso, acontecerão três sorteios em roleta: um por cada extinção / por cada invocação do comportamento **extinguish**. Após cada um destes sorteios, a hiperligação em processo de extinção, recolhe os URLs presentes no recurso vencedor, através do método utilitário **availableUrlsAt**, da **ClassAmUtil**.

O URL substituto do URL a extinguir, é escolhido aleatoriamente dessa lista.

Os resultados são imprevisíveis, podendo apenas ser prognosticados nos termos da tabela anterior. Em todo o caso, se o visitante da página, quiser observar como ficou a comunidade de hiperligações stigmérgicas após a extinção tripla, terá de fazer *reload/refresh* da página.

## 6.11. Resumo

Neste exemplo, a sorte ditou que as três extinções conduzissem a URLs provenientes de três origens diferentes, o que significa que o sorteio em roleta ditou três resultados distintos em outros tantos lançamentos. Se o visitante cujo click desencadeou o trio de extinções, desejar ver os efeitos na página, terá – como nas ocasiões anteriores – de regressar e fazer *reload/refresh* dos conteúdos.

As mudanças são dramáticas. **CS4**, **CS9** e **CS10** extinguiram-se. Os destinos aos quais ligavam foram substituídos por novos destinos, colhidos da lista de recursos dos **cStigh** sobreviventes, na altura.

Nos sorteios em roleta que aconteceram:

- **CS4** sorteou **CS2** para «pai» do seu novo URL: <http://www.formula1.com/news/4965.html> (probabilisticamente, **CS2** era a ligação em melhores condições, com  $P=6/19$ );
- **CS9** sorteou **CS3** e passou a ligar a [http://www.gamasutra.com/php-bin/column\\_index.php?story=8406](http://www.gamasutra.com/php-bin/column_index.php?story=8406) (a probabilidade de eleição de **CS3** era  $P=2/19$ );
- **CS10** sorteou **CS5** ( $P=2/19$ ) e o seu novo URL tornou-se [http://www.arturmarques.com/wordpress/htm/Dancing\\_in\\_the\\_Dark.htm](http://www.arturmarques.com/wordpress/htm/Dancing_in_the_Dark.htm).

Os novos estados de **CS4**, **CS9** e **CS10** são muito diferentes dos estados das restantes hiperligações stigmérgicas. A sua vida é, em termos relativos, muito elevada: 50 pontos, contra 12 da ligação preferida, até ao momento.

O valor de vida inicial para uma ligação substituta só é configurável pela recompilação do sistema. A ideia é não permitir que o autor da página tenha controlo sobre a vida de ligações que tenham nascido pela interacção natural com o STIGH. O autor da página só pode controlar a vitalidade das hiperligações seminais. No exemplo ilustrado, uma vez que a vida máxima para objectos é de 100, é legítimo pensar que o autor pretendeu *não* facilitar a vida dos **cStigh** originais. A vida de 50 para os novos **cStigh** explica-se porque a vida inicial para descendentes está, na versão corrente, a ser computada em metade da vida máxima ( $100/2=50$ ).

As novas ligações também ainda não têm qualquer click, como seria de esperar, para «recém-nascidas».

Com este décimo click, terá ficado ilustrado o funcionar básico do sistema, desde a inicialização até eventos tão violentos como uma extinção tripla.



Figura 16 – Exemplo ilustrado de utilização do STIGH - dez clicks (e três extinções) ocorridos

## 7. Ficheiros de registo de estado

Todos estes acontecimentos foram sendo registados pelas próprias hiperligações stigmérgicas. Na versão corrente do STIGH, é possível visitar remotamente o ficheiro de registo de actividade, de qualquer **cStigh**, a qualquer momento.

Por exemplo, esta sequência ilustrada foi provocada em poucos minutos sobre a página disponível em <http://stigh.org/>, depois de se ter feito *reset* ao sistema, o que significa ter eliminado os registos antigos. O *reset* é provocado por um novo tipo de objecto auxiliar (o simulador de utilizações), introduzido mais adiante.

Como foram feitos 10 clicks, cada **cStigh** tomou nota do seu estado, em 10 linhas, uma por cada um desses eventos. Para consultar o registo de cada objecto, basta conhecer o seu **ID** e o nome da página. Num acesso programático, pode perguntar-se esse nome directamente ao objecto, pelo método público **stateDefaultFileName**.

Para o STIGH de demonstração, os URLs para acesso aos ficheiros de registo de estado estão descritos na Tabela 6.

<b>ID</b>	<b>Ficheiro de registo de estado</b>
CS1	<a href="http://stigh.org/~ Default.aspx CS1 STATE.TXT">http://stigh.org/~ Default.aspx CS1 STATE.TXT</a>
CS2	<a href="http://stigh.org/~ Default.aspx CS2 STATE.TXT">http://stigh.org/~ Default.aspx CS2 STATE.TXT</a>
CS3	<a href="http://stigh.org/~ Default.aspx CS3 STATE.TXT">http://stigh.org/~ Default.aspx CS3 STATE.TXT</a>
CS4	<a href="http://stigh.org/~ Default.aspx CS4 STATE.TXT">http://stigh.org/~ Default.aspx CS4 STATE.TXT</a>
CS5	<a href="http://stigh.org/~ Default.aspx CS5 STATE.TXT">http://stigh.org/~ Default.aspx CS5 STATE.TXT</a>
CS6	<a href="http://stigh.org/~ Default.aspx CS6 STATE.TXT">http://stigh.org/~ Default.aspx CS6 STATE.TXT</a>
CS7	<a href="http://stigh.org/~ Default.aspx CS7 STATE.TXT">http://stigh.org/~ Default.aspx CS7 STATE.TXT</a>
CS8	<a href="http://stigh.org/~ Default.aspx CS8 STATE.TXT">http://stigh.org/~ Default.aspx CS8 STATE.TXT</a>
CS9	<a href="http://stigh.org/~ Default.aspx CS9 STATE.TXT">http://stigh.org/~ Default.aspx CS9 STATE.TXT</a>
CS10	<a href="http://stigh.org/~ Default.aspx CS10 STATE.TXT">http://stigh.org/~ Default.aspx CS10 STATE.TXT</a>

Tabela 6 – Consulta online dos registos de evolução de estado

Eis o ficheiro de registo da evolução do estado de **cStigh CS9**, ao cabo das 10 utilizações ilustradas, na página em que está inserido:

```
CS9 | 9/18/2006 9:03:52 PM | 85.139.98.17 | 1789833183 | Click # | 1 | sUrl | http://imdb.com | dLife | 10 | iMaxLife | 100 | iClicks | 0 | clicks@page | 1 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:04:11 PM | 85.139.98.17 | 1789833183 | Click # | 2 | sUrl | http://imdb.com | dLife | 8 | iMaxLife | 100 | iClicks | 0 | clicks@page | 2 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:04:30 PM | 85.139.98.17 | 1789833183 | Click # | 3 | sUrl | http://imdb.com | dLife | 8 | iMaxLife | 100 | iClicks | 0 | clicks@page | 3 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:04:44 PM | 85.139.98.17 | 1789833183 | Click # | 4 | sUrl | http://imdb.com | dLife | 6 | iMaxLife | 100 | iClicks | 0 | clicks@page | 4 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:04:57 PM | 85.139.98.17 | 1789833183 | Click # | 5 | sUrl | http://imdb.com | dLife | 6 | iMaxLife | 100 | iClicks | 0 | clicks@page | 5 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:05:11 PM | 85.139.98.17 | 1789833183 | Click # | 6 | sUrl | http://imdb.com | dLife | 4 | iMaxLife | 100 | iClicks | 0 | clicks@page | 6 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:05:17 PM | 85.139.98.17 | 1789833183 | Click # | 7 | sUrl | http://imdb.com | dLife | 4 | iMaxLife | 100 | iClicks | 0 | clicks@page | 7 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:05:40 PM | 85.139.98.17 | 1789833183 | Click # | 8 | sUrl | http://imdb.com | dLife | 2 | iMaxLife | 100 | iClicks | 0 | clicks@page | 8 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:05:49 PM | 85.139.98.17 | 1789833183 | Click # | 9 | sUrl | http://imdb.com | dLife | 2 | iMaxLife | 100 | iClicks | 0 | clicks@page | 9 | stighs@page | 10 | iTrigger | 2 |
CS9 | 9/18/2006 9:07:13 PM | 85.139.98.17 | 1789833183 | Click # | 10 | sUrl | http://www.gamasutra.com/php-bin/column\_index.php?story=8406 | dLife | 50 | iMaxLife | 100 | iClicks | 0 | clicks@page | 10 | stighs@page | 10 | iTrigger | 2 |
```

O formato da escrita é:

```
ID | data e hora | IP | Sessão | Click # | ?0 | sUrl | ?1 | dLife | ?2 | iMaxLife | ?3 | iClicks | ?4 | clicks@page | ?5 | stighs@page | ?6 | iTrigger | ?7 |
```

Passa a explicar-se:

- **ID** – é o ID da ligação stigmérgica, conforme a sua declaração na página;
- **data e hora** – são a data e a hora, do lado do servidor, no momento do click;
- **IP** – endereço IP do cliente que fez o click;
- **xxxxxxxxxx** – um nº que identifica univocamente a sessão do cliente, com o sistema;
- **Click #** – frase literal, estática, para explicar o valor que surge à sua direita;
- **?0** – contador absoluto de clicks na página (meramente incremental);
- **sUrl** – frase literal, estática, para explicar o valor que surge à sua direita;
- **?1** – URL de destino, da ligação stigmérgica;
- **dLife** – frase literal, estática, para explicar o valor que surge à sua direita;
- **?2** – vitalidade (apesar de no exemplo ser um inteiro, pode ter parte decimal, como em 1.5);
- **iMaxLife** – frase literal, estática, para explicar o valor que surge à sua direita;
- **?3** – vida máxima, imposta pela configuração do sistema;
- **iClicks** – frase literal, estática, para explicar o valor que surge à sua direita;

- **?4** – contador privado de clicks no objecto;
- **clicks@page** – frase literal, estática, para explicar o valor que surge à sua direita;
- **?5** – contador de clicks na página (pela comunicação do objecto com os seus semelhantes);
- **stighs@page** – frase literal, estática, para explicar o valor que surge à sua direita;
- **?6** – contador de objectos cStigh presentes na página;
- **iTrigger** – frase literal, estática, para explicar o valor que surge à sua direita;
- **?7** – ritmo de degradação do sistema.

Há dados desnecessários, por linha. Embora o IP do visitante seja registado, esse valor nunca é utilizado. Fica a nota de que é muito simples modificar as hiperligações stigmérgicas, para que recolham todo o contexto da ligação com o cliente, o que poderá facilitar aplicações destinadas a medir exaustivamente a utilização da ligação, havendo pois potencial para uma granularidade mais fina, para aplicações de *web metrics* – discute-se esta possibilidade na secção 10.

As expressões literais que prefixam alguns dados no ficheiro, são desnecessárias para quem conheça o formato da escrita. Neste caso pretendeu-se conferir maior legibilidade ao registo.

## 8. Observações

De seguida recorre-se à notação  $STIGH(n)$ , para observações, sobre a aplicação do sistema de hiperligações stigmérgicas, em páginas com  $n$  objecto(s) do tipo **cStigh**, sendo  $n$  um inteiro positivo.

As observações são válidas para a configuração **stigh.cfg** já apresentada = { |, 2, 100 }

### 8.1. Página com um cStigh: STIGH(1)

Numa página com uma só ligação, por não haver alternativas e porque o mecanismo de envelhecimento está apenas indexado à contagem de clicks e não à passagem do tempo, a ligação solitária irá viver «eternamente».

Em rigor não há lugar à stigmérgia, porque não há lugar à comunicação entre objectos.

A ligação existente vai ter uma vitalidade tendente para **iMaxLife**: sendo a única, é a única que pode ser preferida/reforçada, por **getStronger**. Note-se que a ligação também é sujeita a **getWeaker**, a cada **iTrigger=2** clicks.

Assumem-se triviais os comportamentos de reforço (**getStronger**) e de penalização (**getWeaker**), porque não correspondem a mais do que uma soma elementar, considerando que a subtracção é um caso particular da adição.

Aritmética do reforço (**dLife** representa a vitalidade da ligação):

$$dLife \text{ } \beta \text{ } dLife + 2 * iTrigger$$

Aritmética da penalização:

$$dLife \text{ } \beta \text{ } dLife - iTrigger$$

### 8.2. Página com dois cStigh: STIGH(2)

Em páginas com dois **cStigh**, já é possível o fenómeno da extinção (**sparkExtinctions**) de uma das ligações.

A particularidade deste sistema é que a roleta proporcional vai estar organizada numa só fatia, correspondente à única ligação sobrevivente.

A extinção implica a formação de uma roleta proporcional e a realização de um sorteio aleatório. Este fenómeno pode acontecer em todas as páginas com pelo menos dois **cStigh**; ou seja, é um fenómeno que ocorre, sempre da mesma forma, em todos os  $STIGH(n)$ , com  $n \geq 2$ .

### **8.3. Páginas $STIGH(n)$ , $n > 2$**

A única diferença de  $STIGH(n)$ , com  $n > 2$ , relativamente a  $STIGH(2)$ , é que a roleta utilizada no processamento de extinções, tem mais fatias.

O processo de extinção é exactamente o mesmo, como não poderia deixar de ser, perante a implementação do sistema na linguagem formal C#.

A computação da roleta faz-se conforme explicado na secção de «Cálculo de probabilidades e roleta proporcional».

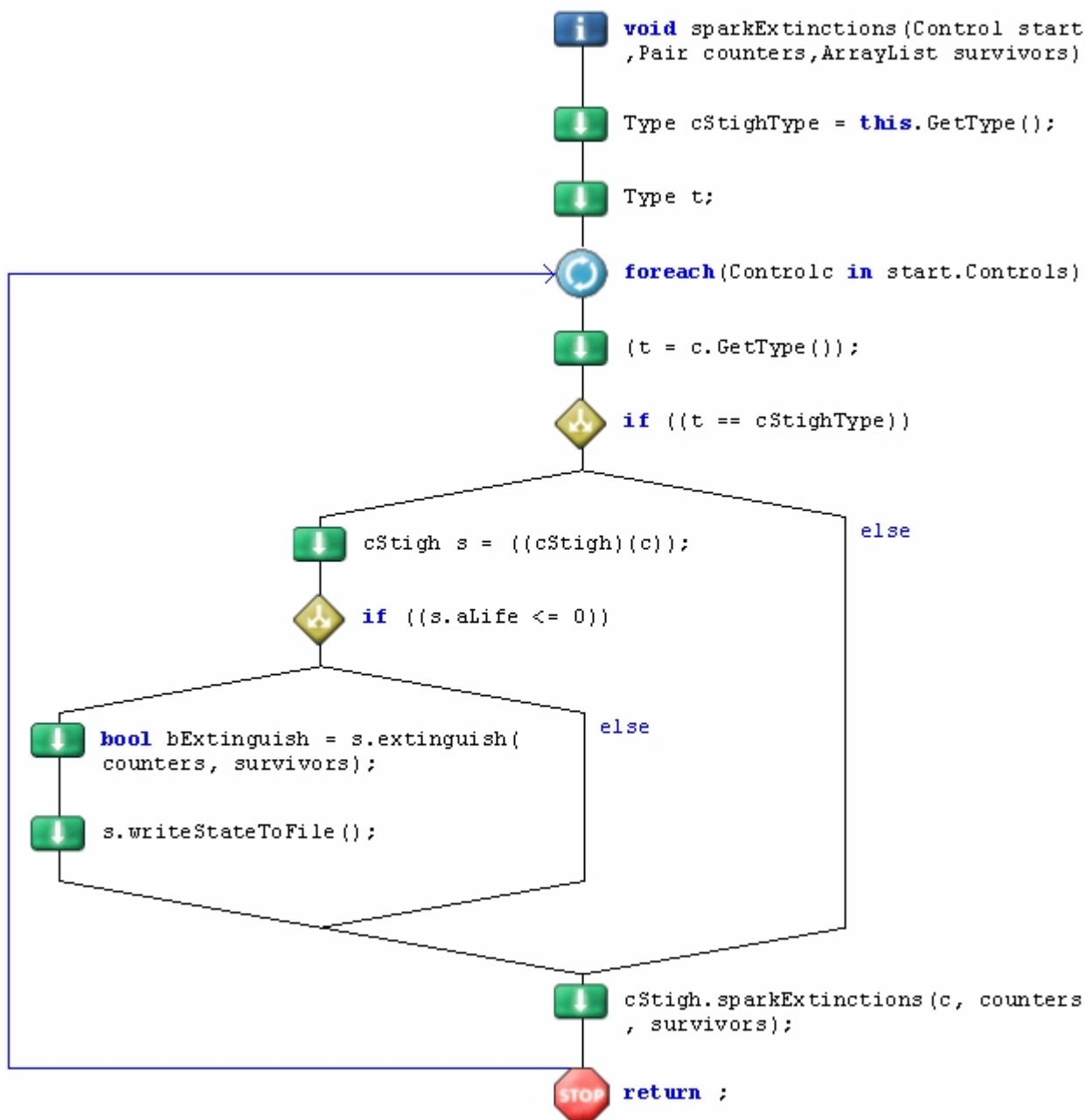


Figura 17 – cStigh – sparkExinctions – Diagrama de fluxo

## 8.4. Satisfação do objectivo

Tanto quanto é possível concluir, pelo documentado nos capítulos 6, 7 e 8, pelas especificações formais e pelos registos colhidos, a abordagem stigmérgica proposta é efectivamente capaz de conferir dinâmica às hiperligações de uma página, de maneira a que evoluam consoante as preferências de navegação dos seus utilizadores.

## 9. Engenharia do sistema

### 9.1. Classes e outras estruturas de informação

Para concretizar o sistema stigmérgico de hiperligações (**STIGH**) numa realidade utilizável em websites, foi feita uma solução, programada na linguagem C#, recorrendo à plataforma Microsoft .NET, e ao ambiente de desenvolvimento Visual Studio.

A plataforma .NET é gratuita e está disponível para diversas configurações de hardware/sistema operativo. Para além de compiladores pela linha de comandos, estão também disponíveis ambientes integrados de desenvolvimento (Visual Studio Express), para as linguagens VB.NET e C#.

A linguagem C# é puramente orientada a objectos, pelo que não é possível *não* construir pelo menos uma classe, qualquer que seja a solução que se pretenda programar.

A classe mais importante da solução que corresponde ao sistema stigmérgico é a classe **cStigh**. A solução está configurada para *não* gerar um executável, mas antes uma biblioteca de classes (**webstigh.dll**), facilmente distribuível.

A biblioteca de classes em causa consiste essencialmente em dois novos controlos (objectos receptivos a eventos), destinados a funcionarem do lado do servidor (*custom server controls*): o controlo mais importante é o que permite hiperligações stigmérgicas (classe **cStigh**); o outro controlo (classe **cStighDemo**) é um auxiliar que permite simular utilizadores do sistema, de forma a facilitar testes com, eventualmente, milhares de eventos de click.

A solução inclui ainda classes utilitárias, que *não* correspondem a controlos. Essas classes (**ClassAmUtil** e **ClassAmRandom**) são repositórios de métodos importantes, relacionados com o processamento de páginas externas ao website (**ClassAmUtil**) e com sorteios aleatórios (**ClassAmRandom**).

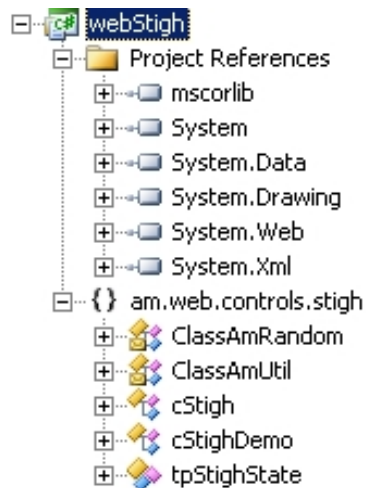


Figura 18 – Class View, no ambiente Visual Studio, para a solução desenvolvida

Do ponto de vista da arquitectura da informação, a solução inclui ainda uma estrutura (**tpStighState**) muito simples, concebida para representar o trio de dados, correspondentes à informação de estado de uma hiperligação stigmérgica (vida, clicks e URL de destino).

Na classe **cStigh**, o membro de dados **state** é do tipo **tpStighState**. A informação de estado tem de estar constantemente preservada.



Figura 19 – Estrutura tpStighState, para o estado interno de cada objecto cStigh

## 9.2. Declaração de hiperligações stigmérgicas

Para que um website possa servir hiperligações stigmérgicas – objectos instâncias da classe **cStigh** ou, simplesmente, **STIGHs** – basta que (1) esteja instalada a tecnologia .NET e (2) aloje o ficheiro **webstigh.dll** na sua directoria *\bin*.

Cumpridos estes dois requisitos, o website em causa pode servir páginas que incluam STIGHs (e **cStighDemos**).

Para os autores de conteúdos, a inclusão de STIGHs em páginas exige apenas que se aprenda a utilização de uma nova marca, a utilizar na linguagem de marcação.

Por exemplo, o troço de código seguinte, descreve um STIGH:

```
<cc1:cStigh ID="CStigh1" aLife="10" aUrl="http://arturmarques.com"
runat="server"></cc1:cStigh>
```

Neste exemplo, a declaração é inaugurada pelo elemento **cc1:cStigh**. O encerramento faz-se explicitamente em **</cc1:cStigh>**. O prefixo (cc1) é configurável.

O atributo **ID** deve ser único por página, e distingue o objecto que se está a declarar, de todos os outros. Se se cometer o erro de declarar dois ou mais objectos com o mesmo ID, a própria tecnologia .NET não processa a página e acusa o erro.

Os STIGHs também podem ser construídos visualmente, desde que utilizando ambientes de desenvolvimento compatíveis, como o Microsoft Visual Studio.

A construção visual pode ser feita com apenas um duplo-click no ícone que representa os objectos **cStigh**, na barra de ferramentas (*toolbox*). O objecto será automaticamente declarado na página, com valores por defeito e com um **ID** garantidamente único.

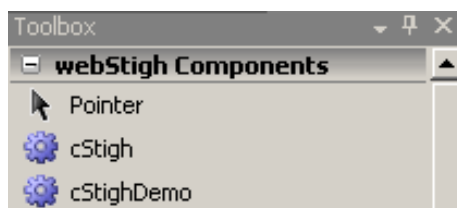


Figura 20 – Ferramentas para a inserção de ligações stigmérgicas em páginas, no ambiente Visual Studio

O atributo **aLife** determina a vida inicial da ligação stigmérgica. Valores elevados facilitam a preservação do objecto. Valores baixos facilitam a sua extinção. A extinção é um evento a que os STIGHs estão sujeitos, quando se lhes esgota a vitalidade – nessa altura, acontece-lhes a mudança do recurso de destino (ver o atributo **aUrl**).

Se, na declaração do objecto, procurar-se atribuir-lhe valores «absurdos» para a vida, como valores negativos ou valores acima do limite estabelecido como o máximo (membro de dados **iMaxLife**), a atribuição será desconsiderada e o objecto será construído com um valor de vida inicial, que será metade do máximo (**iMaxLife/2**).

O valor máximo **iMaxLife** é configurável, conforme referido na secção 9.4.

O atributo **aUrl** determina o recurso de destino. É o equivalente ao atributo **href**, para o elemento **a** em HTML, que marca as hiperligações comuns, na WWW – o maior sistema de hipertexto.

O atributo **runat** indica que a entidade declarada é um objecto a processar do lado do servidor. É muito importante recordar que uma hiperligação comum é uma entidade que corre do lado do cliente: quando alguém, utilizando um browser, faz click sobre uma hiperligação, o que acontece de seguida está totalmente sob controlo do browser. Com o STIGH, há necessidade de sinalizar ao servidor que o click aconteceu naquele instante, pelo que o evento correspondente tem que ser «furtado» ao browser e feito processar do lado do servidor.

Perante o *status quo* de processamento de hiperligações do lado do cliente, o STIGH enfrentou um desafio técnico interessante.

Para conseguir-se hiperligações que correm do lado do servidor, fez-se um pouco de ilusionismo: um objecto **cStigh** é desenhado (*rendered*) na página como se fosse uma hiperligação (método **RenderContents**), mas comporta-se como um botão. Do ponto de vista da hierarquia de classes, a classe **cStigh** é derivada (herdeira) de **LinkButton**, que corresponde, na plataforma .NET, precisamente a esta necessidade.

Para além dos atributos aqui sumariamente introduzidos, um objecto **cStigh** pode ser declarado de forma mais sucinta e também de forma mais complexa, com menos ou mais atributos.

Os atributos que estão disponíveis para serem utilizados na declaração, são determinados pelas chamadas propriedades (*accessors*) que – de forma simplista – são funções sem lista de parâmetros, que fazem a leitura e/ou a alteração de membros de dados da classe, correspondendo assim a um mecanismo de escudo contra acessos directos.

### 9.3. Prefixo declarativo

O prefixo (*cc1 / custom control 1*) é configurável, aplicando-se a todos os objectos **cStigh**, na página ou em todo o website, dependendo de onde venha a ser configurado.

Para configurar o prefixo por página, basta que, no início, a página reclame o prefixo com uma sintaxe como:

```
<%@ Register Assembly="webStigh" Namespace="am.web.controls.stigh" TagPrefix="cc1" %>
```

Para configurar o prefixo a nível de todo o website, basta que no ficheiro XML **web.config**, que configura um website capaz de servir .NET, se escreva:

```
<add tagPrefix="cc1" namespace="am.web.controls.stigh"></add>
```

Este elemento tem de ser escrito abaixo da hierarquia **system.web\pages\controls**. Onde está “cc1” deve ficar a frase que se pretende para prefixo.

Estes são detalhes de administração de websites .NET, pelo que não aprofunda esta informação.

## 9.4. Ficheiro de configuração: stigh.cfg

O ficheiro **stigh.cfg** deverá ser um ficheiro de texto, com exactamente três linhas, presente na raiz do website.

As linhas determinam (1) a frase separadora que se utiliza nas escritas que aparecem em *tooltip* e na *statusbar* do browser do cliente; (2) a cadência de envelhecimento das hiperligações stigmérgicas na página, medida em quantidade de eventos de click; e (3) a vida máxima para qualquer objecto **cStigh**.

Segue-se um exemplo de ficheiro **stigh.cfg** e uma explicação mais detalhada das suas consequências.

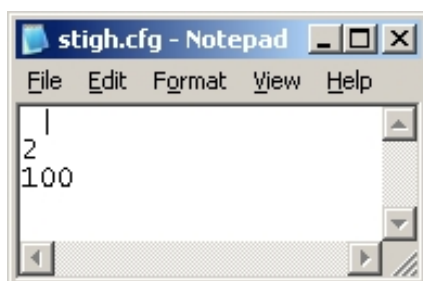


Figura 21 – Ficheiro de configuração stigh.cfg

A primeira linha do ficheiro consiste numa barra vertical (“ | ”), com um espaço em branco à esquerda e outro à direita. O que esta linha determina é a frase separadora, utilizada nas *tooltip* e *statusbar* do browser, quando se pede informação sobre uma hiperligação stigmérgica, estacionando o ponteiro do rato em cima do objecto. A frase que aparece é determinada pelo método **stighInfo**.

**sSeparator** estabelece a frase separadora por defeito (“ \* ”), quando *não* existe ficheiro de configuração.

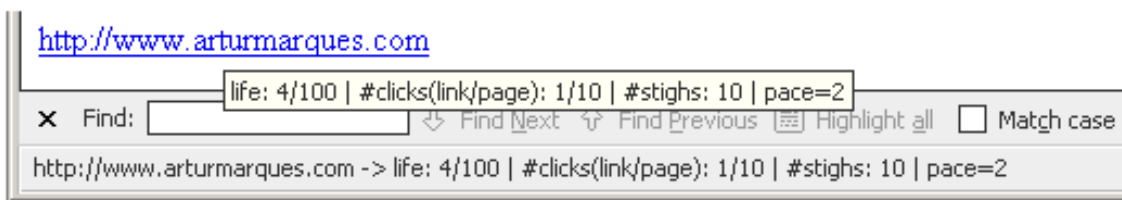


Figura 22 – Informação sobre a ligação stigmérgica em *tooltip* e na *statusbar*

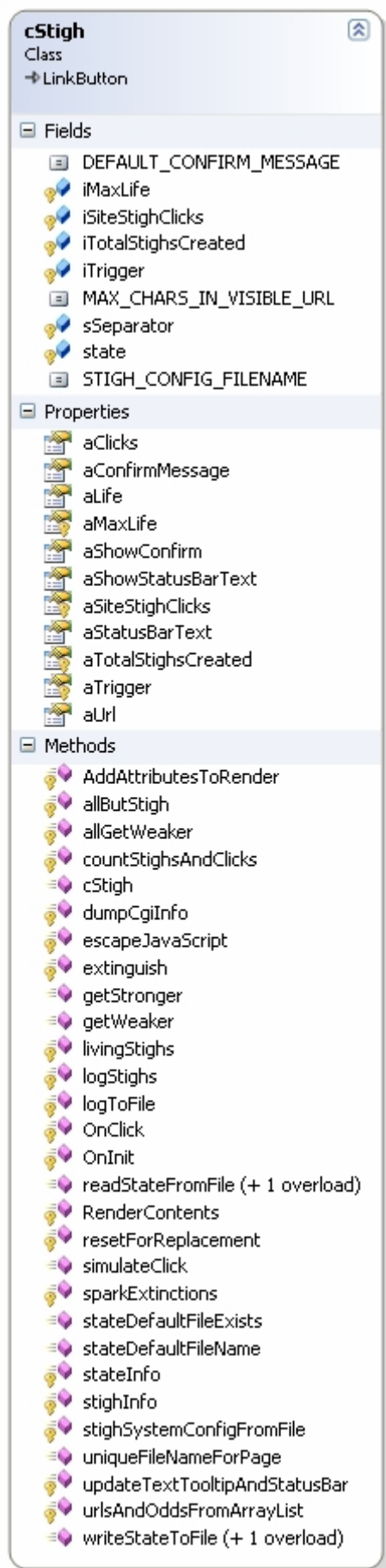


Figura 23 – Classe cStigh (representação parcial do seu diagrama de classes)

## 9.5. Classe cStigh – Discussão, a partir das constantes

No diagrama de classes à esquerda – debitado automaticamente pelo ambiente de desenvolvimento Microsoft Visual Studio –, ilustram-se os membros de dados (*fields*), as propriedades (*properties*), e os métodos (*methods*) da classe **cStigh**.

Está assinalado que se trata de uma classe derivada de **LinkButton**.

Os membros de dados incluem três constantes públicas:

- (1) **DEFAULT\_CONFIRM\_MESSAGE**,
- (2) **STIGH\_CONFIG\_FILENAME** e
- (3) **MAX\_CHARS\_IN\_VISIBLE\_URL**.

Para compreender a primeira constante é necessário saber que um objecto **cStigh** pode, opcionalmente, perguntar ao utilizador que nele faz um click, se pretende realmente navegar até ao destino da ligação.

Este comportamento é controlado pela propriedade lógica **aShowConfirm**. Por defeito, esta propriedade é *false*, pelo que a hiperligação stigmérgica *não* pedirá confirmação de navegação, quando for sujeita ao evento de click.

Se um autor de conteúdos desejar que um STIGH em particular peça confirmação de navegação, através de uma caixa de diálogo elementar, deve fazer a declaração adequada:

<... *aShowConfirm*="true" ...>

Estando a opção **aShowConfirm** activa, a mensagem com a qual o utilizador será questionado é configurável, pela propriedade **aConfirmMessage**.

Assim, por exemplo, para declarar-se um STIGH que pergunte “ir?”, quando alguém lhe faz click, basta escrever:

```
< ... aShowConfirm="true" aConfirmMessage="ir?" ...>
```

Quando se omite a valoração de **aConfirmMessage**, é utilizada a frase, em língua inglesa, “*Follow the link?*”, que é o valor da constante simbólica **DEFAULT\_CONFIRM\_MESSAGE**.

Embora seja possível configurar a questão de confirmação, não é possível configurar o que surge na barra de título da caixa de diálogo. Isto acontece porque a caixa de diálogo é conseguida pela injeção do código Javascript *return confirm(questão);* no browser do cliente, sendo que a função *confirm* não é configurável quanto ao título.

Browsers diferentes poderão apresentar títulos diferentes, para esta caixa de diálogo. Por exemplo, o browser Mozilla Firefox faz título com o URL do website; o Microsoft Internet Explorer, utiliza a frase “Microsoft Internet Explorer”...



Figura 24 – Exemplo de caixa de diálogo para confirmação da navegação

A constante pública **STIGH\_CONFIG\_FILENAME** determina o nome do ficheiro que configura os parâmetros mais gerais do sistema stigmérico de hiperligações. Esta constante só é alterável pela recompilação da solução pelo que, na prática, na versão corrente, está imposto o nome **stigh.cfg**.

### 9.5.1. Apresentação da informação de estado: *stighInfo*

Quando se pede a um objecto **cStigh** o comportamento (a execução do método) **stighInfo**, o objecto responde com uma frase que descreve o seu estado interno. Essa frase tem quatro partes: (1) informação sobre a vida do objecto, na forma *vida-actual/vida-máxima*; (2) informação sobre os

clicks, na forma *clicks-no-objecto/clicks-na-página*; (3) informação sobre a quantidade de objectos **cStigh** já criados pelo servidor *na-página*; e (4) de quantos em quantos clicks na página é que se desencadeia o envelhecimento dessa comunidade de STIGHs.

Na *tooltip* da Figura 22, pode ler-se que (1) o objecto em causa está com uma vida de 4, num máximo de 100; (2) aconteceram 1 click nele próprio e 10 na página; (3) existem 10 **cStigh** na página; e (4) a página desencadeia o envelhecimento das suas ligações stigmérgicas, de 2 em 2 clicks.

Note-se que a separação das quatro partes da frase é feita com a barra vertical (“ | ”). É precisamente para decidir a estética desta separação que serve a primeira linha do ficheiro **stigh.cfg**, para a configuração do sistema.

Isto aplica-se também ao que surge na *statusbar* do browser: a zona de informações, normalmente residente no canto inferior esquerdo da janela do browser.

Todavia, note-se que a informação na *statusbar* está prefixada pelo URL de destino do objecto **cStigh** – essa é a única diferença de apresentação entre a informação de estado via *tooltip* e via *statusbar*<sup>1</sup>.

Se o texto correspondente ao URL exigir mais do que **MAX\_CHARS\_IN\_VISIBLE\_URL** caracteres (92, por defeito), a sua frase é encurtada até esse valor e sufixada com “...”, para acusar o corte.

A segunda linha do ficheiro (2) está a impor que todas as hiperligações stigmérgicas, na página, sejam envelhecidas a cada dois clicks. Se se pretender a quebra de vitalidade de todos, a cada selecção singular, este valor deverá ser 1. Se se pretender outra cadência, este valor deverá reflectir essa intenção. Esta cadência valora o membro de dados **iTrigger**, através da propriedade de interface **aTrigger**.

---

<sup>1</sup> Nota: por defeito, o browser Mozilla Firefox impede modificações à sua *statusbar*. É por isso que a informação do STIGH é replicada em *tooltip*. Para forçar o Firefox a permitir alterações à *statusbar*, o utilizador do Firefox deve (1) escrever o endereço *about:config* na barra de endereços; e (2) valorar a propriedade *dom.disable\_window\_status\_change* a *false*. Este comportamento do Firefox não tem qualquer relação com o STIGH.

Valores elevados de **iTrigger** correspondem a envelhecimentos espaçados ou raros. Valores baixos correspondem a um envelhecimento frequente, ou rápido. De forma a contemplar esta frequência/raridade, a intensidade do envelhecimento está-lhe indexada. Se o envelhecimento é «lento», quando acontece, acontece com «violência»; se o envelhecimento é frequente, quando acontecer, não é tão agressivo.

Na versão corrente do STIGH, quando uma ligação stigmérgica é seleccionada, a sua vida é reforçada numa certa *intensidade* que está calculada como sendo o dobro de **iTrigger** ( $2 * \mathbf{iTrigger}$ ). A lógica é que a *intensidade* da recompensa deve ser tanto maior, quanto mais dilatada for a cadência estabelecida por **iTrigger** (valores elevados de **iTrigger** computam resultados maiores, pela expressão referida).

O envelhecimento está a fazer-se com uma intensidade que é metade da utilizada no reforço ( $intensidade/2$ ). Outras lógicas e outros valores de reforço/penalização poderiam ter sido utilizados. Valores diferentes determinam comportamentos de sistema diferentes, como extinções mais céleres ou promoções mais permanentes, mas este estudo *não* objectiva dissertar sobre os resultados dessas diferentes combinações e suas consequências, tendo antes fixado o duo de valores referido, para recompensa/penalização. Considerou-se esta opção como adequada para focar a investigação na arquitectura, materialização e documentação da stigmergia de hiperligações (com uma configuração estável).

A terceira e última linha do ficheiro de configuração do sistema (100), Figura 21, estabelece a vida máxima para qualquer objecto **cStigh**. O valor fica reflectido em **iMaxLife**. No exemplo, o valor é 100.

Este valor não só estabelece um tecto que impede valorações abusivas, mas também um base de cálculo para a vida de novas ligações: por defeito, uma nova hiperligação stigmérgica «nasce» com a vida de  $\mathbf{iMaxLife}/2$ .

Na versão corrente, os autores de conteúdos só conseguem controlar a vida inicial das hiperligações «semente», que são aquelas que efectivamente declaram na página que escrevem.

Logo que o sistema esteja em funcionamento, ele (sistema) torna-se como «selvagem» e perde as características «domésticas» que assumia em tempo de concepção (*design time*). Por exemplo, após as eventuais extinções das hiperligações «semeadas» durante a autoria, as novas ligações dependem totalmente do universo de sobreviventes no momento e da sua rede de recursos (fora do controlo do autor da página), «nascendo» com uma vida que é sempre metade da máxima (também fora do controlo do autor da página).

O valor **iMaxLife** não deve ser alterado, depois de iniciado o sistema, porque isso corresponde a criar contextos diferentes para as ligações que nasceram antes e depois da alteração, o que pode

desvirtuar a captura de preferências dos utilizadores do sistema (do website). A única forma de alterar este valor é editando directamente o ficheiro de configuração – a propriedade que lhe está associada (**aMaxLife**) só permite a consulta (e não a alteração) do valor.

### 9.5.2. Discussão, baseada nos membros de dados *protected*

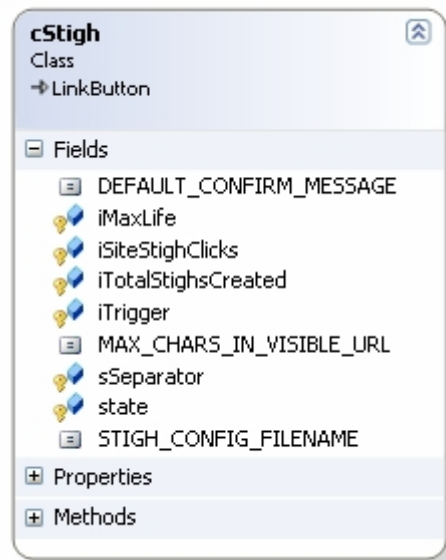


Figura 25 – `cStigh` - *Fields* / membros de dados

Com excepção das constantes, todos os membros de dados da classe **cStigh** são *protected*. Isto significa que só estão acessíveis dentro da própria classe e em classes derivadas, que especializem a hiperligação stigmérgica.

Alguns dos membros de dados são também *static*. Isto significa que são partilhados por todas as instâncias da classe, pelo que a memória que lhes corresponde é comum – se um objecto altera o valor, todos sentem a alteração. O desenvolvimento da versão corrente do STIGH não teve qualquer preocupação de optimização no consumo de recursos – foi orientada à demonstração do conceito –, mas a utilização de membros *static* e o que isso realmente significa é importante para compreender os algoritmos de descoberta que os objectos **cStigh** realizam, por exemplo para saberem quantos objectos do seu tipo existem na página.

O membro de dados *static* **iSiteStighClicks** é um contador de quantos eventos de click, sobre objectos **cStigh**, já aconteceram em todo o website. Este valor é simplesmente incrementado a cada click, em qualquer página, a qualquer momento, por qualquer visitante. A propriedade de interface com este campo é **aSiteStighClicks** e só permite o acesso de leitura.

Este valor *não* tem qualquer importância para o algoritmo do comportamento de um **cStigh** individual, nem para o comportamento do conjunto das hiperligações stigmérgicas. Recorde-se que para um sistema stigmérgico (de hiperligações) só deve importar o espaço (de dados) imediatamente próximo: no caso, a página que publica a ligação (e não todo o website).

É positiva a irrelevância deste valor, por dois motivos, um estrutural, outro meramente técnico.

(1) Se a manutenção deste valor determinasse alguma coisa no comportamento individual das hiperligações stigmérgicas, isso sugeriria que OU comportamento do sistema estava na sombra de uma entidade controladora supra-STIGH, que «olharia para baixo e faria contagens», OU que cada objecto teria de auto-descobrir o valor, procurando no «mundo inteiro» (todo o website), o que romperia com a simplicidade do comportamento stigmérgico dos insectos sociais, que foram fonte inspiradora deste trabalho, porque estar-se-ia perante «criaturas» que analisariam muito mais do que o seu «espaço imediato» e perante resultados de conjunto que não seriam verdadeiramente emergentes.

(2) A manutenção de valores *static* em classes de objectos destinadas a serem suportadas em ambiente servidor de http, está sujeita à robustez desse serviço terceiro, ao longo do tempo.

No STIGH, os membros de dados *static* são efectivamente utilizados para a comunicação entre ligações stigmérgicas, mas *apenas* no espaço de dados imediato; isto é, apenas entre objectos presentes na mesma página.

O membro de dados *static* **iTotalStighsCreated** é um contador de quantos objectos **cStigh** já foram construídos, pelo construtor da classe. A sua actualização faz-se a cada chamada do construtor: o método para a instanciação de **cStighs**. Por exemplo, sempre que uma página que publica duas ligações stigmérgicas é carregada, em alguma sessão, acontecem duas construções. Note-se que não existe uma classe central controladora a fazer a contagem ou outros serviços: a contabilidade correcta resulta exclusivamente do comportamento individual de cada ligação stigmérgica e da natureza da comunicação por memória partilhada (*static*).

Mais uma vez, este valor *não* tem qualquer importância para o algoritmo do comportamento de um **cStigh** individual, nem para o comportamento do conjunto das hiperligações stigmérgicas.

O valor foi mantido na contabilidade do sistema e também na informação retornada por **stighInfo**, para destacar fenómenos que poderão surpreender. Por exemplo, um utilizador pode achar estranho ler (em *tooltip* ou na *statusbar*, ou nos ficheiros de registo que são mantidos) que existem 2 links stigmérgicos na página e 20 links construídos, no total, quando «um segundo antes» existiam 2 links na página e 6 links ao todo... O fenómeno explica-se porque outros utilizadores do sistema

terão requisitado a mesma página (setes vezes, no total:  $7 \times 2 = 14$  ;  $14 + 6 = 20$ ), a partir das suas sessões particulares, durante esse «um segundo».

O método **stighInfo** só reporta estas informações, tipicamente supérfluas, na versão de *debug* do STIGH (uma versão que debita mais informação técnica interna, para auxiliar na detecção de eventuais problemas). A versão de demonstração que está publicada, é uma versão de *release* que omite o desnecessário.

O membro de dados *static* **iTrigger** estabelece de quantos em quantos clicks envelhecem todas as hiperligações stigmérgicas, como consequência *natural* do facto de estarem «vivas». Por outras palavras, **iTrigger** é o relógio do sistema. Uma hiperligação stigmérgica «velha» não é uma ligação que esteja publicada há muito tempo, mas uma ligação que tenha sobrevivido a muitos eventos de selecção (a muitos clicks). Estas duas medidas (tempo e quantidade de eventos de selecção) não têm necessariamente qualquer relação de proporcionalidade: em teoria, um objecto **cStigh** pode estar publicado há anos, mas ser «jovem», porque nesse período a sua página esteve «intocada»; por outro lado, um objecto **cStigh** com poucos dias pode ser um «ancião», se nesse período a sua página tiver sido muitíssimo utilizada e ele tiver conquistado muitas preferências de utilização (muitos clicks).

Um valor representado em membros de dados *static* é preservado durante toda a vida do ambiente que suporta a classe. Num programa executável «convencional», isso é durante toda a execução do programa.

No caso particular da classe **cStigh**, concebida para enriquecer servidores web com novos objectos (a hiperligação stigmérgica e o simulador de utilização), os valores *static* serão preservados enquanto que o servidor de http, com a plataforma .NET e a biblioteca **webstigh.dll** instaladas, estiver online.

Para que a possibilidade de problemas com o servidor de http não comprometa os dados fulcrais – a informação de estado de cada objecto **cStigh** –, o membro de dados **state** é preservado em memória não volátil (ficheiro).

O membro de dados **state** é uma estrutura que agrupa três dados: a vida (**dLife**), os clicks (**iClicks**) e o URL (**sUrl**) do objecto.

Note-se que pela soma dos clicks de todos os objectos **cStigh** na página, é possível computar a quantidade de eventos de selecção no «espaço imediato»: a página corrente. Essa informação é importante para o comportamento de cada ligação stigmérgica individual, porque quando esse

contador (de clicks em objectos da página) for múltiplo de **iTrigger**, desencadeia-se o envelhecimento da comunidade.

Por ser um valor computável, a quantidade de clicks na página não é um valor crítico, que tenha que ser representado em memória não volátil, como se faz com **state**.

Não havendo entidades superiores, a computação deste valor só é possível pela colaboração de todos os **cStigh** na página. O problema é menos simples do que pode parecer, porque as ligações stigmérgicas são construídas à medida que a página vai sendo carregada pelo cliente.

Imagine-se uma página com três ligações stigmérgicas L1, L2 e L3; quando L1 «nasce», L2 e L3 ainda não foram construídas pelo que L1 contabiliza somente 1 **cStigh** (ele próprio); quando L2 «nasce», já encontra 2 **cStighs** (ele próprio e L1); só L3 é que descobre a página «completa», já com os outros objectos (L1 e L2).

Ao comunicarem essa contagem por memória (*static*) comum, os objectos **cStigh** conseguem tornar irrelevante o momento do seu carregamento no lado do cliente.

### 9.5.3. Preservação da informação de estado: *writeStateToFile*

A informação de estado que importa manter, para cada hiperligação stigmérgica, é um trio de valores, representado no membro de dados **state**. Ao contrário da generalidade dos problemas de preservação do estado, que só se colocam ao nível da sessão – como, por exemplo, preservar o nome de uma pessoa num formulário web –, a informação de cada STIGH tem que ser encarada numa perspectiva mais vasta, porque as acções de utilizadores em sessões diferentes têm *sempre* efeito no sistema, desde logo, nas consequências das acções de outros utilizadores.

Um cenário bem ilustrativo de influências cruzadas é aquele em que um visitante de uma página com STIGHs, encontra uma ligação no limiar da extinção. O visitante poderá decidir seguir essa ligação, embora a sua vida residual sugira que não tem conquistado a preferência dos outros visitantes, mas quando resolve fazê-lo, fica surpreso ao ser remetido para um destino que não tem nada a ver com aquele que, *aparentemente* seleccionou.

O que se passou foi que aquilo que o utilizador via no browser, deixou de corresponder à realidade presente do sistema, uma vez que outro(s) visitante(s) terão extinguido a ligação em causa, *aparentemente* seleccionada, ao preteri-la, nas suas sessões. Quando o utilizador *iludido* resolve seguir a hiperligação, faz um pedido ao servidor, que identifica, pela propriedade **ID**, o mesmo STIGH, mas responde com o novo URL que, no acto da extinção, entretanto tomou o lugar daquele que era o esperado.

Tal como em Astronomia é frequente olhar-se para o que não já existe (estrelas a anos-luz de distância, que já pereceram, mas cuja luz outrora emitida só agora nos chega), também no STIGH há potencial para, facilmente, o browser ser uma janela para o passado.

A informação de estado de cada STIGH tem pois que ser preservada acima do nível de sessão. A solução técnica encontrada, na versão corrente, é o simples recurso a ficheiros.

Cada STIGH tem um ficheiro de estado associado, escrito na raiz do website. O nome do ficheiro tem a forma **nomeDaPágina\_ID.CFG**. Por exemplo, para um STIGH com o ID “S1”, presente na página P1, o ficheiro de estado é P1\_S1.CFG.

Cada ficheiro de estado é um ficheiro de texto, com extensão **.CFG** e cinco linhas: a primeira para o **ID** do objecto, a segunda para a vida (**dLife**), a terceira para o contador de clicks (**iClicks**), a quarta para o URL de destino (**sUrl**), e a quinta para o texto associado ao URL (**Text**).

Um objecto **cStigh** escreve o seu estado em disco, quando se lhe pede o comportamento **writeStateToFile**. Este método tem dois *overloads* (duas declarações com listas de parâmetros diferentes), podendo ser invocado sem argumentos – ocasião em que faz a escrita para a raiz do website, respeitando o formato já mencionado – ou com indicação explícita do ficheiro onde há-de escrever.

#### 9.5.4. Propriedades públicas

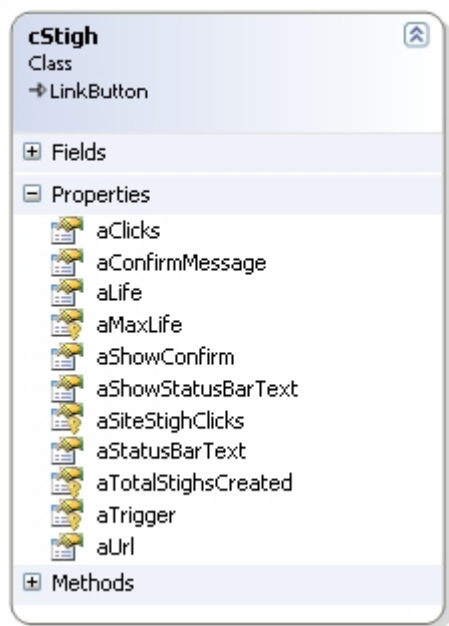


Figura 26 – cStigh - *Properties* / propriedades

As propriedades declaradas como públicas são sete: **aClicks**, **aShowConfirm**, **aConfirmMessage**, **aLife**, **aUrl**, **aShowStatusBarText** e **aStatusBarText**.

Estas propriedades, exactamente com estes nomes, correspondem a atributos, que podem aparecer na declaração da hiperligação stigmérgica, na página.

**aClicks** permite (apenas) a consulta de **iClicks** – o contador que cada hiperligação stigmérgica mantém, de quantas vezes já foi seleccionada.

**aLife** permite a consulta e a valoração de **dLife** – a vida inicial do objecto.

**aUrl** permite a consulta e a valoração de **sUrl** – o URL de destino do objecto.

**aConfirmMessage**, determina a mensagem que surge na caixa de diálogo, para confirmação da navegação. Este valor *não* corresponde a nenhum membro de dados, sendo mantido numa região de memória, conhecida por **ViewState**, pensada para a preservação de informação visível, do lado do cliente, entre sucessivas submissões da página ao servidor.

Existe uma fragilidade na utilização de **ViewState** para informação crítica, em *custom server controls*. Como controlos, como a hiperligação stigmérgica, destinam-se a serem distribuídos e utilizados em diferentes websites, estão sujeitos às diferentes configurações que encontrarem. Acontece que a memória **ViewState** pode ser desactivada pela administração do servidor, ou mesmo pelo autor da página, comprometendo o funcionamento dos objectos que nela confiarem. Alguns administradores e autores desactivam **ViewState**, porque a preservação dos valores em **ViewState** faz-se, na prática, pela injeção de campos escondidos na página que o cliente recebe, tornando a página mais extensa (mais «pesada»).

No caso de **cStigh**, se **ViewState** estiver desactivado, as hiperligações stigmérgicas *não* permitirão (1) que se configure a opção de mostrar, ou não, a caixa de confirmação; (2) que se configure a

mensagem que surge na caixa de confirmação; (3) que se active/desactive a utilização da *statusbar* do browser; e (4) que se configure a mensagem a aparece na *statusbar* do browser.

Os valores de **aShowConfirm**, **aConfirmMessage**, **aShowStatusBarText** e **aStatusBarText**, seriam então os por defeito e ignorariam as valorações eventualmente tentadas na declaração.

Uma vez que estas propriedades são essencialmente acessórias, optou-se por não representá-las em mecanismos mais permanentes, como seriam **ControlState** (disponível na plataforma .NET v2), ou pelo recurso a ficheiros, tal como se faz para a informação de estado.

Um pouco como os contadores absolutos de construções e de clicks – já discutidos –, estas propriedades mantidas em **ViewState**, são «luxos» dispensáveis.

#### *9.5.5. Propriedades protected*

**aMaxLife**, **aTrigger**, **aSiteStighClicks** e **aTotalStighsCreated**, são propriedades protegidas que *não* podem ser usadas como atributos.

**aMaxLife** e **aTrigger** permitem a consulta da vida máxima e do ritmo do sistema, tomando valores pelo ficheiro de configuração.

**aSiteStighClicks** e **aTotalStighsCreated** permitem a consulta de contadores (supérfluos) para o comportamento do sistema, de quantos clicks já aconteceram e de quantos objectos já foram construídos, em todo o website.

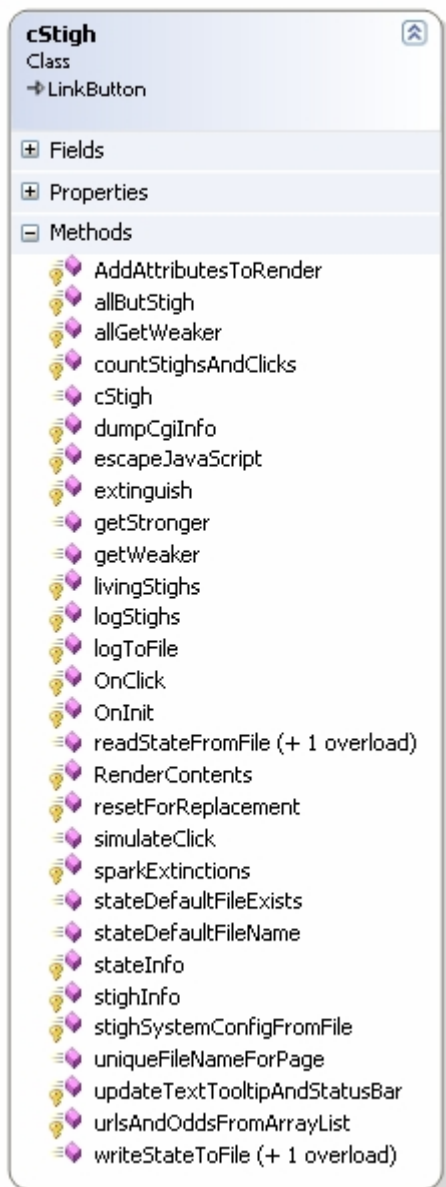


Figura 27 – cStigh – Métodos

Os métodos correspondem aos comportamentos que uma ligação stigmérgica sabe desempenhar.

### 9.5.6. Métodos públicos

Os métodos públicos correspondem aos comportamentos que qualquer entidade pode pedir a uma hiperligação stigmérgica.

Os métodos *protected* só podem ser invocados dentro da classe ou por classes herdeiras.

Os comportamentos públicos de cada **cStigh** são:

- **getStronger**,
- **getWeaker**,
- **writeStateToFile**,
- **readStateFromFile**,
- **uniqueFileNameForPage**,
- **stateDefaultFileName**,
- **simulateClick**.

O construtor também é um comportamento público.

Todos os outros métodos são *protected* e cumprem funções de um nível de abstracção inferior.

**getStronger** permite a uma ligação stigmérgica ganhar vitalidade. A quantidade de vitalidade é parâmetro do método (e não um valor fixo).

Este comando deve ser enviado, sempre que um **cStigh** merecer uma recompensa, por ter sido seleccionado na página. O algoritmo do método é cauteloso, impedindo vitalidades acima de **iMaxLife** e abaixo de zero – esta última seria teoricamente possível, se a recompensa enviada fosse negativa.

**getWeaker** permite a uma ligação stigmérgica perder vitalidade. A quantidade de vitalidade é parâmetro do método (e não um valor fixo).

Este comando executa-se a cada **iTrigger** eventos de click, na página, e corresponde ao envelhecimento natural do sistema. O algoritmo do método é cauteloso e impede valores fora do intervalo [0, **iMaxLife**].

**getStronger** e **getWeaker** alteram o estado interno da hiperligação stigmérgica, pelo que solicitam a preservação do novo estado, de imediato, invocando o comportamento **writeStateToFile**.

**writeStateToFile** escreve, em ficheiro de texto, o estado do objecto. Este método pode ser chamado de duas formas: sem argumento ou referindo o nome do ficheiro. A forma de maior nível de abstracção é a primeira: nesse caso, a escrita acontece para a raiz do website, para um ficheiro cujo nome identifica univocamente a página e a hiperligação stigmérgica (ver a secção «**Preservação da informação de estado: writeStateToFile**») – esse nome de ficheiro pode ser obtido a qualquer momento, pelo método **stateDefaultFileName**.

**writeStateToFile** é cauteloso e captura excepções de *runtime* que possam acontecer, como a falta de espaço no suporte, violações de permissões, falhas de hardware e quaisquer outras não previstas. Infelizmente, nesses cenários catastróficos, embora o sistema permaneça em funcionamento, ao não conseguir memória para actualizar o estado da hiperligação, o seu comportamento deixará de reflectir a realidade da utilização da página.

É de notar que, a acontecerem, estes cenários dever-se-ão a falhas em níveis físicos e/ou lógicos que o STIGH não controla. Por isso, na medida do que está ao seu alcance, considera-se que o STIGH é tolerante e robusto, perante as falhas mencionadas.

**readStateFromFile** lê, de ficheiro de texto, o estado do objecto. Este método pode ser chamado de duas formas: sem argumento, ou referindo o nome do ficheiro. Ao ser invocado sem argumento, procura recuperar o estado do objecto a partir do ficheiro cujo nome pode ser obtido via **stateDefaultFileName**.

**readStateFromFile** é tolerante a falhas, como impedimentos no acesso ao ficheiro, por qualquer problema, desde permissões a avarias de hardware. Todavia, nesses cenários catastróficos, impedido de recuperar o estado do objecto, o STIGH terá de atribuir-lhe o estado por defeito, pelo que até à regularização do problema, o sistema vai permanecer cristalizado, sem reflectir a real utilização da página.

**readStateFromFile** só é invocado na inicialização do controlo, no método **OnInit**, chamado automaticamente pelo ambiente .NET, durante o carregamento da página, no browser do cliente.

**uniqueFileNameForPage** é um método utilitário, ao qual uma hiperligação stigmérgica responde com um «nome único», para a página onde está alojada. Um «nome único» é uma frase que identifica a página univocamente. Foi opção do algoritmo deste método, produzir identificadores de muito alto nível, pelo que a frase devolvida é uma expressão legível que descreve a localização da página e o seu nome, no sistema de ficheiros do servidor.

Por exemplo, para uma página P1, situada numa directoria D1, o nome será D1\_P1.

**stateDefaultFileName** é um método que responde o nome do ficheiro onde a hiperligação stigmérgica está a preservar a sua informação de estado.

**simulateClick** é um método que permite forjar eventos de click, sobre a ligação. Este comportamento só está disponível nas versões de demonstração do STIGH, de forma a facilitar simulações de utilização.

A par da classe **cStigh**, que permite hiperligações stigmérgicas, foi desenvolvida uma classe – **cStighDemo** – que permite incluir nas páginas simuladores/demonstradores de utilização. É graças a esta classe que é possível conseguir registos de actividade com milhares de clicks, que sugerem a robustez do sistema.

O método **simulateClick** *não* deve ser público nas versões finais do STIGH, porque isso viabiliza que pessoas conhecedoras do sistema, possam falsear a utilização da página. Por exemplo, o seguinte troço de código (em C#) desencadeia 1000 clicks na ligação stigmérgica com ID “id1”, na página P1:

```
stighObject = (cStigh)ClassAmUtil.getControlWithId(P1,"id1");  
for (int i=0; i<1000; i++) stighObject.simulateClick();
```

O construtor de **cStigh** também é um método público. A particularidade no desenvolvimento de *custom server controls* é que *não* tem interesse disponibilizar múltiplos construtores.

Embora o código da classe **cStigh** pudesse ter múltiplos construtores (por exemplo, poder-se-ia pensar num construtor que recebesse a vida inicial do objecto como parâmetro), só o denominado construtor por defeito (aquele que não tem quaisquer parâmetros) é que será utilizado. Isto é uma limitação modesta da plataforma .NET – modesta, porque a inicialização dos membros de dados está prevista pela utilização de propriedades, em combinação com atributos, na declaração do objecto.

### 9.5.7. Métodos *protected*

Alguns dos métodos *protected* em **cStigh** são herdados de **LinkButton**: um **cStigh** é uma especialização de um **LinkButton**.

Enquanto especializações de **LinkButton**, as ligações stigmérgicas exibem comportamentos próprios, como **getWeaker** e **getStronger**, mas também modificam comportamentos herdados.

#### 9.5.8. Métodos herdados, especializados

São herdados e especializados os métodos:

- **AddAttributesToRender**,
- **RenderContents**,
- **OnClick**,
- **OnInit**.

#### 9.5.9. Métodos para o desenho de um cStigh, na página

O desenho (*render*) de um controlo/objecto, na página do cliente, faz-se pelo comportamento **RenderContents**. Mas, antes de **RenderContents** actuar, o objecto pode/deve injectar atributos na *tag* que o descreve na página: esses atributos terão efeitos.

A injeção de atributos antes do desenho, tem que ser feita em **AddAttributesToRender**.

No caso de **cStigh**, existem atributos que, para resultarem, têm que ser injectados em **AddAttributesToRender**. Esses atributos são **aShowConfirm** e **aConfirmMessage** – determinantes para a caixa de diálogo de confirmação da navegação –, e a **aShowStatusBarText** e **aStatusBarText**, que ditam o que aparece na *statusbar* do browser.

A caixa de diálogo de confirmação, é conseguida pelo acrescento do atributo **onclick**, devidamente valorado, na lista de atributos da hiperligação.

Por exemplo, injecta-se: `onclick="javascript:return confirm('ir?');"`

A injeção faz-se em HTML, no elemento **a**, para *hyperlinks* convencionais. Quem pedir para ver o código fonte da página (com objectos **cStigh**), só vai encontrar construções HTML.

Por exemplo, simplificando, à declaração (do lado do servidor)

```
<cc1:cStigh ID="idl" aUrl="http://site.com" aConfirmMessage="ir?"></cc1:cStigh>
```

vai corresponder o HTML (do lado do cliente)

```
<a href="http://site.com" onclick="javascript:return confirm('ir?');"
href="javascript:__doPostBack(id1','')>site.com</a>
```

Esta transformação do que está declarado do lado do servidor em conteúdo consumível pelo cliente, é característico da tecnologia .NET (e de outras, como PHP, por exemplo).

**RenderContents** é o método que desenha o objecto na página, com os atributos que tiverem sido acrescentados em **AddAttributesToRender**.

No caso de **cStigh**, o método foi alterado relativamente ao método da base (portanto o método em **LinkButton**), porque optou-se por limitar o tamanho do texto das hiperligações, a um máximo de **MAX\_CHARS\_IN\_VISIBLE\_URL** caracteres.

Muito simplesmente, a versão especializada de **RenderContents**, em **cStigh**, limita-se a cortar a escrita de URLs, a partir da posição mencionada. Para assinalar o corte, a escrita termina com "...". O cliente pode continuar a ver o URL completo, na *statusbar* do browser.

#### 9.5.10. Métodos relacionados com o ciclo de vida do objecto, em .NET

**OnInit** e **OnClick** são refinados em **cStigh**, relativamente às versões herdadas de **LinkButton**. Isso é especialmente evidente para **OnClick**, uma vez que uma hiperligação stigmérgica é original na sua reacção a eventos de selecção.

Em **OnInit** só acontecem duas operações: (1) da primeira vez que a ligação é carregada, escreve-se em ficheiro o seu estado inicial; das vezes seguintes, o estado é recuperado/lido do ficheiro; (2) lê-se a configuração da própria página/sistema.

**OnClick** é mais complexo. Em rigor, o algoritmo de **OnClick** é o algoritmo que locomove todo o sistema stigmérgico.

#### 9.5.11. Resposta ao evento de click: OnClick

Ao ser seleccionada, uma hiperligação stigmérgica começa por olhar para o seu espaço imediato (a página), para descobrir quantas outras ligações **cStigh** existem; durante essa descoberta também questiona as suas semelhantes sobre quantas vezes já foram seleccionadas, podendo assim –

facilmente – computar o total de clicks na página. Tudo isto é feito pelo método **countStighsAndClicks**.

Por ter sido preferida, a ligação sente-se reforçada, na sua vitalidade. Esse reforço acontece pelo comportamento **getStronger**. A intensidade do reforço é configurável. Na versão corrente, o reforço é o dobro do ritmo do sistema ( $2 * \mathbf{iTrigger}$ ), pelo que em cadências dilatadas (**iTrigger** elevados), a recompensa é maior, num esforço de consideração da menor/menor «raridade» do envelhecimento sistémico, que também conduzirá a penalizações de vitalidade menos/mais violentas, para as ligações preteridas: a penalização corrente é metade do reforço.

**iTrigger** estabelece pois de quantos em quantos eventos de click, os **cStigh** da página deverão enfraquecer. Matematicamente, basta verificar se o total de clicks na página é múltiplo de **iTrigger**. Ao verificar-se esta condição, a hiperligação comunica a todas as semelhantes no seu espaço imediato (a página) que devem enfraquecer: esse pedido faz-se pelo método **allGetWeaker**.

Após o enfraquecimento colectivo, põe-se a possibilidade de alguns **cStigh** terem ficado com vida nula. Nesse caso, devem extinguir-se. Esta verificação e a eventual extinção, faz-se pelo comportamento **sparkExinctions**, discutido em detalhe na secção sobre extinções.

Por fim, há que cumprir duas formalidades: (1) tendo o estado do objecto **cStigh** sido alterado, pelo click, há que preservá-lo, por **writeStateToFile**; e (2) há que remeter o visitante para o URL, como se tivesse seleccionado uma hiperligação convencional.

Em configuração de demonstração, é possível que o STIGH *não* remeta os utilizadores para o URL de destino. A vantagem disso, para demonstrações, é que o visitante pode assim seguir a actualização da página do sistema (e não perder a vista da página, substituída pelo URL de destino).

#### *9.5.12. Métodos de suporte ao funcionamento de cada hiperligação stigmérgica*

Os métodos referidos como públicos e invocados no processamento de um click, são os métodos de maior nível de abstracção, para o funcionamento do STIGH. A generalidade destes métodos, funciona sobre uma camada lógica de suporte.

O desenvolvimento do STIGH foi quase sempre feito em abordagem *top-down*, o que significa que os sucessivos problemas foram sendo decompostos em problemas menores, na lógica da orientação a objectos; ou seja, comportamentos mais abstractos foram motivando comportamentos mais específicos.

Por exemplo:

- **sparkExinctions** confia que cada **cStigh** sabe extinguir-se; isto é, confia no método **extinguish**;
- **extinguish** confia em diversos métodos, relacionados com a eleição do sobrevivente que, na página, há-de fornecer um sucessor;
- os registos (*logs*) de actividade que vão sendo mantidos por comportamentos como **logStighs**, confiam em métodos como **dumpCgiInfo**, para obtenção de informações como o IP do cliente.

### 9.5.13. A extinção: *sparkExinctions* + *extinguish*

De todos os comportamentos, a extinção é o mais «dramático» e, tecnicamente, o mais complexo. Isto acontece devido à componente exógena do processo: quando um **cStigh** se extingue, ele deve ser substituído por uma ligação proveniente de uma das páginas externas, às quais um dos sobreviventes está a ligar-se. Isto implica processar recursos externos e identificar-lhes hiperligações. Várias situações podem correr mal: para além do recurso poder já não estar disponível, pode tratar-se de um recurso «estéril», no sentido de vazio de ligações...

Os dois métodos que cada **cStigh** sabe executar, relacionados com extinções, são **sparkExinctions** e **extinguish**.

**sparkExinctions** funciona como um pedido colectivo ao sistema, para que as hiperligações stigmérgicas averiguem a sua vitalidade e se extingam, se for caso disso: será caso disse, se tiverem vida nula. Um **cStigh** lança este pedido quando é seleccionado e é momento de desencadear o envelhecimento sistemático. Do ponto de vista da implementação, isto não é mais do que um ciclo por todos os objectos **cStigh** na página: sempre que um dos objectos tiver vida zero, o seu comportamento **extinguish** será pedido.

Quando se ordena a uma hiperligação stigmérgica que se extinga, compete-lhe arranjar um sucessor, de entre as ligações presentes nas páginas às quais ligam os **cStigh** sobreviventes. Um objecto sabe identificar os sobreviventes, no momento da sua extinção, pelo método **livingStighs**.

O **cStigh** sobrevivente que deve fornecer o sucessor, é sorteado, respeitando as preferências implicitamente capturadas pelo **STIGH** até ao momento; ou seja, os **cStigh** com mais vida deverão ter maior probabilidade de eleição.

O sorteio a realizar é pois um sorteio em roleta proporcional: as maiores fatias devem corresponder aos objectos mais vivos. Por exemplo, um objecto  $o_1$  cuja vida seja a dobro de outro  $o_2$ , deve ter o dobro da probabilidade de ser eleito –  $p(o_1)=2*p(o_2)$  – pelo que a sua fatia na roleta deve ser duas vezes maior.

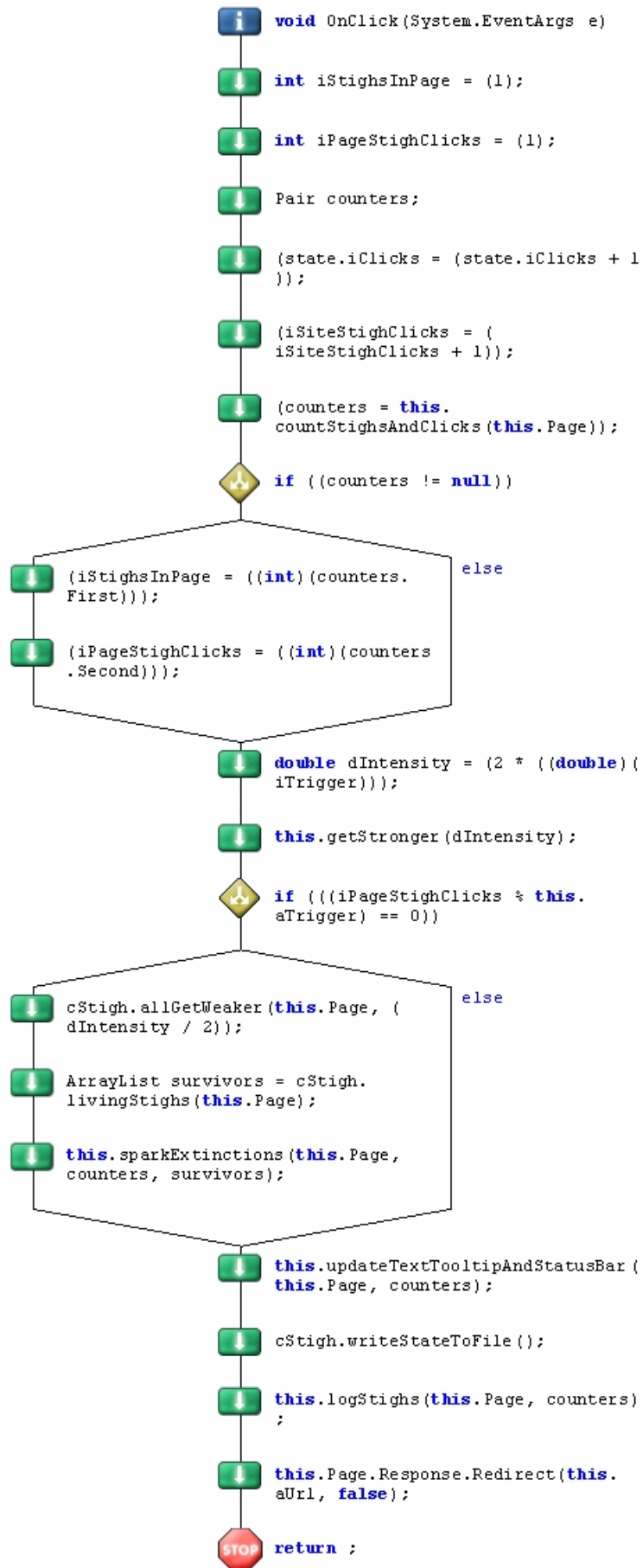


Figura 28 – cStigh – OnClick – Diagrama de fluxo

## 9.6. Classes auxiliares

A classe **ClassAmRandom** foi desenvolvida, precisamente para auxiliar na realização de sorteios em roleta proporcional e sorteios aleatórios.

A classe **ClassAmUtil** é uma contentora de métodos utilitários, relacionados com a componente exógena do STIGH: é esta classe que contém todas as ferramentas necessárias para processar páginas externas.

### 9.6.1. Classe *ClassAmRandom*

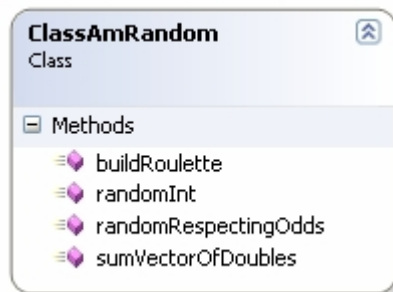


Figura 29 – ClassAmRandom

A classe **ClassAmRandom** só disponibiliza quatro métodos, todos públicos e *static*. Um método *static* é um método que é partilhado por todas as instâncias da classe, pelo que funciona mais como uma função ou procedimento, pronto a ser utilizado a qualquer momento, em qualquer lado, sem necessidade de instanciar algum objecto.

**buildRoulette** é uma função que contrói uma roleta proporcional. Esta função recebe as probabilidades que têm que ser reflectidas nas fatias, e devolve a roleta, na forma de uma família de intervalos no espaço  $[0,1]$ . Por exemplo, se a função receber o conjunto de probabilidades  $\{0.5, 0.5\}$ , vai devolver um conjunto de dois intervalos:  $\{ [0, 0.5], ]0.5,1] \}$ . Esses intervalos são as fatias. No caso do exemplo, as fatias são idênticas, porque descrevem eventos equiprováveis.

**randomInt** é uma função que devolve um número inteiro aleatório, no intervalo que for especificado pela sua lista de dois parâmetros. Por exemplo, pode pedir-se a **randomInt** um inteiro em  $[10, 100]$ .

**randomRespectingOdds** é um função que devolve um inteiro, sorteado por recurso a uma roleta proporcional. Esta função só precisa de receber um conjunto de probabilidades, porque depois (1) constrói a roleta correspondente, chamando **buildRoulette**; (2) sorteia um número real em [0, 1]; (3) situa esse número real numa das fatias da roleta, retornando o número (inteiro) da fatia.

**randomRespectingOdds** é pois extremamente útil, para sortear o **cStigh** sobrevivente que vai fornecer a hiperligação substituta da extinta: basta invocá-la com a lista das probabilidades para cada **cStigh**. Note-se que um **cStigh** sem vida não causa problemas ao sorteio – ao ter vida zero, terá probabilidade zero, pelo que não terá direito a fatia...

**sumVectorOfDoubles** é uma função auxiliar, apenas utilizada para verificar a correcção matemática de listas de probabilidades: toda a lista tem que somar 1.

## 9.6.2. Classe ClassAmUtil

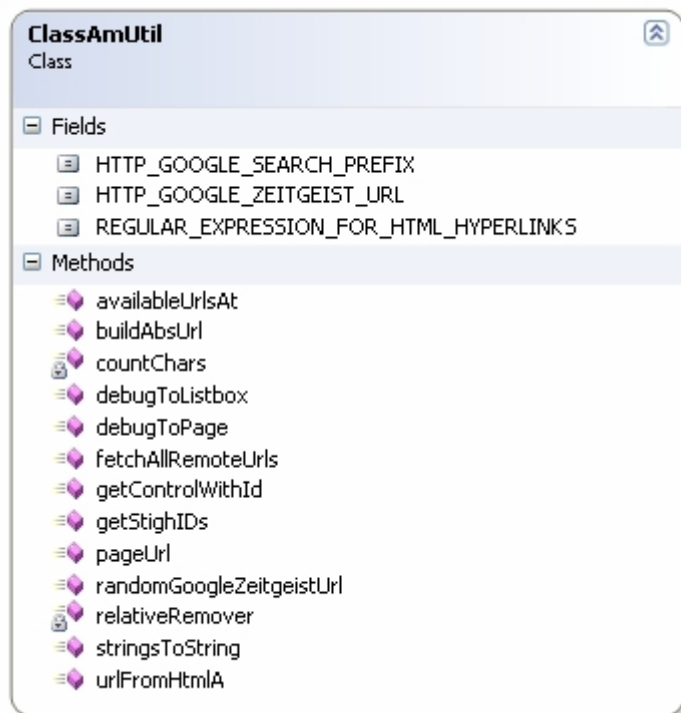


Figura 30 – ClassAmUtil

Esta classe é um repositório de métodos públicos utilitários, que podem ser usados como funções ou procedimentos, porque não necessitam de uma instância. **ClassAmUtil** foi desenvolvida para suportar a componente exógena do **STIGH**: a generalidade das suas ferramentas serve, directa ou indirectamente, para processar qualquer página HTML, identificar-lhe hiperligações (elementos HTML **a**), e ajudar a contornar situações em que uma página é «fechada» ou «estéril», providenciando uma solução que permite sempre produzir uma ligação, recorrendo ao serviço *Zeitgeist* do Google (<http://www.google.com/press/zeitgeist.html>).

**availableUrlsAt** é uma função que, recebida uma página HTML, responde com a lista de todas as hiperligações presentes nessa página. As hiperligações são identificadas pela presença de elementos HTML **a**, no código fonte. Por exemplo, em `<a href="http://y.com">y.com</a>`, descobre-se a ligação <http://y.com>.

As hiperligações que estejam presentes na forma de URLs relativos, são transformadas em URLs absolutos, pelo método **buildAbsUrl**. Por exemplo, de `<a href="../y.htm">y.htm</a>`, pode deduzir-se <http://y.com/y.htm>, se a página estiver um nível hierárquico abaixo da raiz do website.

**availableUrlsAt** está no nível de abstracção mais elevado, da sua classe. Para desempenhar, acaba por recorrer a auxiliares, principalmente **fetchAllRemoteUrls**. Esta última tem um algoritmo muito

elegante, que se limita a aplicar uma expressão regular, para a detecção de hiperligações, na página pretendida. Uma expressão regular é como que uma regra de produção de uma gramática, sendo capaz de gerar certas sequências de símbolos.

A expressão regular que o STIGH utiliza para a detecção de hiperligações é:

```
<a[\s]+[^\>]*?href[\s]?=[\s\'"]+(.*?)[\'"]+. *?>([^\<]+|. *?)?</a>
```

Fez-se corresponder a esta expressão, a constante pública

**REGULAR\_EXPRESSION\_FOR\_HTML\_HYPERLINKS**, de forma a facilitar a sua utilização programática.

Esta expressão encontra todas as sequências de símbolos começadas por **<a** e terminadas em **</a>**, desde que refiram algum recurso (atributo **href**). Não é importante que a frase refira o atributo **href** logo de início.

Como seria de esperar, o que importa na frase, é o URL propriamente dito. Para a extracção do URL, do interior das frases que a expressão regular identifica, utiliza-se **urlFromHtmlA**.

As outras constantes públicas declaradas nesta classe, são

**HTTP\_GOOGLE\_SEARCH\_PREFIX** ( <http://www.google.com/search?q=> ), e

**HTTP\_GOOGLE\_ZEITGEIST\_URL** ( <http://www.google.com/press/zeitgeist.html> ).

**pageUrl** é um método que responde o URL absoluto de qualquer objecto do tipo *Page*, na plataforma .NET. Este método é útil para que uma hiperligação stigmérgica descubra o URL absoluto da *Page* em que está alojada.

**debugToListbox** é um utilitário, sem interesse para a compreensão do STIGH, escrito para facilitar o acompanhamento do funcionamento do sistema, nas suas primeiras fases. É de notar que depurar (fazer o *debug*) de um novo objecto para servidores web é profundamente diferente de fazer *debug* de uma aplicação tradicional, porque o ambiente do objecto é, afinal, uma página carregada num browser.

**getControlWithId** é uma função que retorna o objecto que tenha o **ID** que se peça, caso exista, em determinada página.

**getStighIDs** retorna a lista dos identificadores de objectos do tipo **cStigh**, em qualquer página. Este método é explorado pelos simuladores de actividade no STIGH: um simulador (1) identifica os

**cStigh** na página; (2) obtém a referência programática de qualquer um desses objectos, através de **getControlWithId**; e, a partir de aí, (3) pode comandar o objecto como entender, como enviando-lhe ordens de **simulateClick**.

**randomGoogleZeitgeistUrl** é um método de muito alto nível, sem parâmetros, que retorna um URL aleatório, sorteado a partir de procuras feitas, sobre as palavras mais procuradas, na semana anterior à corrente, via Google.

A palavra *zeitgeist* tem origem etimológica nas palavras alemãs *zeit* (tempo) e *geist* (espírito), significando o «clima intelectual, moral e cultural de uma era». Na prática, o Google *Zeitgeist* é uma página HTML, com as frases mais procuradas na semana anterior.

O algoritmo desenvolvido para a função **randomGoogleZeitgeistUrl** é muito simples: (1) faz-se a chamada **availableUrlsAt (HTTP\_GOOGLE\_ZEITGEIST\_URL)**, que retorna os URLs correspondentes às procuras das expressões mais populares via Google; (2) sorteia-se, aleatoriamente, um desses URLs; e termina-se (3) sorteando, aleatoriamente, um dos resultados da procura.

**countChars**, **relativeRemover** e **stringsToString**, são utilitários, sem interesse para a compreensão do STIGH, que servem para operações sobre frases.

### 9.6.3. Classe *cStighDemo*

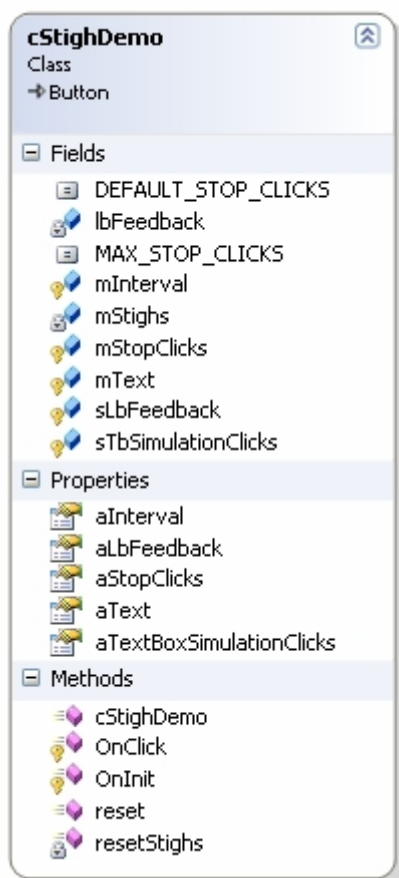


Figura 31 – `cStighDemo`

Esta classe corresponde a um novo controlo/objecto, que pode ser desenhado numa página em que existam hiperligações stigmérgicas. Uma vez que exista um objecto `cStighDemo` na página, pode pedir-se-lhe que faça um qualquer número de selecções sobre a comunidade de `cStighs`.

Na sua versão corrente, este objecto é muito limitado e intrusivo: (1) só faz selecções aleatórias e (2) exige a presença de uma caixa de texto e de um botão, onde um utilizador da página possa, respectivamente, escrever quantos clicks deseja, e dar início à sequência de selecções.

O foco da versão actual do STIGH é a classe `cStigh`, correspondente à hiperligação stigmérgica. Esta outra classe de *custom server control* mereceu a atenção suficiente para permitir o teste com as limitações enunciadas.

Ainda assim, `cStighDemo` é um primeiro passo no sentido de utilizadores artificiais, que – em trabalho futuro – poderão auxiliar na compreensão do comportamento emergente do sistema, em múltiplas configurações.

Na versão corrente, os objectos `cStighDemo` só servem para pôr à prova a robustez mecânica do STIGH, ao longo de milhares de utilizações, que seriam um tédio, a serem feitas manualmente.

Um `cStighDemo` só tem dois comportamentos: (1) com `OnClick`, desencadeia a simulação; (2) com `reset`, apaga a informação de estado de todos os `cStigh` da página, repondo as ligações stigmérgicas na sua configuração original. Internamente, `reset` recorre a `resetStighs` – é este último que itera pelas hiperligações e lhes destrói o ficheiro de preservação de estado.

A simulação pode intervalar os clicks: o valor desse intervalo, medido em milésimos de segundo, está acessível pela propriedade `aInterval` (correspondente ao membro de dados `mInterval`).

A quantidade de clicks está acessível através de **aStopClicks** (correspondente ao membro de dados **mStopClicks**).

Um **cStighDemo** informa sobre que objectos vai actuando. Essa informação é escrita numa *listbox*. A identificação da *listbox* está acessível em **aLbFeedback** (o membro de dados correspondente é **mLbFeedback**).

A identificação da caixa de texto, através da qual se espera que um utilizador da página, requisite certo número de clicks, está acessível através de **aTextBoxSimulationClicks** (o membro de dados correspondente é **sTbSimulationClicks**).

Um **cStighDemo** não deixa de ser um botão (classe herdeira de **Button**), com efeitos muito particulares. O texto que aparece no botão está acessível por **aText** (correspondendo ao membro de dados **mText**).

O membro de dados mais interessante num simulador de clicks em hiperligações stigmérgicas, é **mStighs**, que é a colecção dos identificadores de **cStighs**, presentes na página. Como todos os outros membros de dados não constantes, este dado está protegido (*protected*). A diferença é que, desta feita, *não* existem propriedades que permitam aceder-lhe.

Isto significa que a informação em causa é descoberta pelo próprio simulador, mas não está disponível, para consulta directa. A única forma de conhecer os **cStigh** que o simulador identificou, é através da *listbox* associada, pois quando o simulador se inicializa (**OnInit**), escreve aí os achados.

## 10. Sumário e conclusão

### 10.1. Sumário

Nesta investigação propõe-se um novo objecto de Hipertexto – a hiperligação stigmérgica – desde a ideia em abstracto, à sua concretização na forma de software distribuível.

Numa leitura linear, este documento começa (1) pelo conceito de stigmérgia, (2) enquadra filosoficamente o estudo, (3) apresenta o ciclo de vida de uma hiperligação stigmérgica, (4) uma modelação do sistema, (5) outras investigações sobre hiperligações, e (6) ilustra o funcionamento de um sistema de hiperligações stigmérgicas, tornado possível por uma solução de software designada de STIGH (STIGmérgia de Hiperligações), escrita na linguagem formal C#.

Os capítulos 7 a 9 discutem (7) as diferentes formas possíveis de consulta dos registos de actividade de cada hiperligação particular, (8) casos gerais de sistemas de hiperligações stigmérgicas, e (9) a engenharia do STIGH.

### 10.2. Limitações

Estão identificadas as seguintes **limitações intrínsecas** ao sistema: (1) não é feito qualquer esforço para impedir que visitantes anormalmente frequentes e repetitivos, possam enviar as preferências capturadas; (2) uma ligação stigmérgica não é uma ligação HTML convencional, por mais que o pareça, podendo isso causar frustrações de utilização; e (3) no processamento de hiperligações remotas, só se considera o atributo correspondente ao endereço associado.

A propósito da segunda limitação, está identificada uma situação frustrante, na implementação actual: não resulta a funcionalidade de «*open in a new window*», do browser, para uma ligação stigmérgica, porque o pedido teria que seguir até ao servidor e todos os browsers tratam essa operação localmente.

O facto do sistema exigir a plataforma .NET do lado do servidor, não é tanto uma limitação da sua natureza, mas mais uma restrição à sua adopção generalizada.

Quanto à limitação referida em (3), poderia facilmente processar-se outros dados, como o texto que embebe o URL, mas considerou-se preferível trabalhar nesse sentido, num enquadramento de investigação diferente, como a melhoria da usabilidade das hiperligações HTML actuais, conforme destacado no primeiro tópico de trabalho futuro.

### 10.3. Trabalho Futuro

Referem-se quatro direcções de estudo, possíveis a partir do STIGH corrente.

#### 10.3.1. Melhorias na usabilidade de hiperligações

O conceito corrente de hiperligação está de tal forma entranhado na generalidade dos utilizadores da Internet, que é lhes é difícil identificar melhorias e até formas alternativas de interligar recursos.

Algumas das limitações presentes são a (1) unidireccionalidade, (2) o destino não verificado, e (3) questões de relevância, reputação e confiança (Leuf 2006).

A hiperligação vigente é unidireccional: aponta para um recurso, mas não sabe referir *quais*, ou tão pouco *quantos* são os apontadores para si própria. A quantidade de apontadores para um destino é uma métrica relevante – sugestiva da «qualidade» do destino – que é explorada por serviços como procuras, via Google. Assim, indirecta e (apenas) aproximadamente, para (apenas) alguns destinos, é possível contar quantos são os chamados *backlinks*, ou ligações para uma página de origem. Seria muito interessante que fosse da natureza da hiperligação fornecer essa informação.

A forma de fazê-lo poderia passar por um novo serviço de contagem, contactado aquando de uma utilização da hiperligação, aquando de um click. Afinal, hoje, quando um link é seguido, a aplicação cliente começa por recorrer ao serviço de DNS, para descobrir o IP do servidor que aloja o conteúdo, ao que se segue, tipicamente, a invocação do método GET, do protocolo HTTP, para acesso a uma representação adequada do destino. Poderia, após uma consulta DNS, fazer-se uma consulta a esse hipotético serviço de contabilização de apontadores, que faria uma incrementação, no caso do referenciador ainda não estar identificado.

Esta abordagem obrigaria à reescrita de todos os clientes de hiperligações, para que acessem ao novo serviço, aquando do evento de click num *hyperlink*.

Uma alternativa, que dispensaria a reescrita dos clientes, seria trabalhar do lado do servidor, na resposta ao evento de selecção: ou seja, começar por fazer aquilo que cada hiperligação stigmérgica já faz! Identifica-se aqui um trabalho futuro, com base na classe **cStigh** actual.

A não verificação do destino, significa que as aplicações clientes, remetem para o destino de uma hiperligação, sem terem informação sobre a sua (não) existência e/ou sobre a sua (não) disponibilidade. O destino pode ter sido removido, pode nunca ter existido, ou pode simplesmente estar *off-line*.

Mais uma vez, a arquitectura STIGH poderia ser explorada, pelo menos no sentido de inquirir sobre a (não) existência do recurso.

Questões de relevância do recurso ligado, e de reputação e/ou confiança do seu provedor, autor e/ou editor, são exemplos de questões prementes, pelo número crescente de conteúdos publicados na Internet, todos tratados exactamente por igual, pelo objecto de hiperligação HTML corrente.

A criação de mecanismos de recolha de reacções aos conteúdos, é necessária para «avaliá-los» e depois explorar essa avaliação de alguma forma. A hiperligação stigmérgica deste estudo é um mecanismo, transparente, para identificar preferências, que pode ser utilizado como base de desenvolvimento para outras identificações.

Hiperligações que enderecem, total ou parcialmente, as limitações (1), (2) e (3), têm potencial para melhorar a experiência dos utilizadores, exibindo, nesse sentido, melhor usabilidade.

Nesta perspectiva, a hiperligação stigmérgica parece constituir também um contributo no sentido de uma melhor usabilidade.

### 10.3.2. Aprofundamento do conhecimento da stigmergia de hiperligações

Este estudo fez-se sobre uma configuração muito particular do sistema: degradação a cada dois clicks (**iTrigger**=2), reforços no valor de  $2 \cdot \mathbf{iTrigger}$  pontos de vitalidade, e penalizações no valor de **iTrigger**.

Estes valores influem no ritmo de envelhecimento da página, na violência do envelhecimento, e no alcance dos reforços da vitalidade. Dissertou-se superficialmente sobre essa influência: um **iTrigger** elevado corresponde a uma degradação de baixa frequência; valores de vitalidade baixos e/ou penalizações elevadas facilitam a extinção... mas não se formalizou um sistema explicativo do impacto de *quaisquer* valores para essas variáveis.

Entretanto, a classe **cStighDemo**, correspondente a um simulador de clicks, tem margem para muitos melhoramentos e especializações. Na sua versão actual, permite objectos que comportam-se de forma totalmente aleatória, servindo para testar a robustez do STIGH, em situações de «stress», como muitas selecções imprevisíveis, em pouco tempo.

Se fosse feita a especialização de **cStighDemo** no sentido de classes de utilizadores com preferências conhecidas e comportamentos bem modelados, então o duo {**cStigh**, **cStighDemo** e derivadas} poderia tornar-se num ambiente para o estudo acelerado da WWW, contribuinte para a produção de respostas a questões como «*que texto há-de ter a hiperligação, com destino D, para melhor atrair pessoas do segmento S?*»...

Atente-se que especializar **cStighDemo** no sentido ambicionado, é um problema semelhante ao do ovo e da galinha: concretizar a especialização implica já conhecer as preferências e os comportamentos, que são exactamente o tipo de informação que se extrai da resposta à pergunta... Logo, é a pergunta que ajuda a construir uma especialização de **cStighDemo**, ou é **cStighDemo** que ajuda a responder à pergunta?

Este dilema sugere a utilidade que poderão ter processos de auto-descoberta e o sensato que é não excluir nenhuma abordagem, *a priori*. Estas questões em particular e a investigação na WWW em geral, apelam à multidisciplinaridade.

### *10.3.3. Possibilidade no sentido de Inteligência Artificial em objectos de hipertexto*

O conceito ilustrado pelo STIGH é, no fundo, a stigmergia de uma comunidade de entidades de hipertexto, para a emergência de maior utilidade para o utilizador, no sentido em que foi discutida. Isto fez-se através de um novo objecto – a ligação stigmérgica – com comportamentos muito simples: penalização e recompensa.

É estimulante reflectir sobre outros novos objectos de hipertexto, que poderiam ser arquitectados, com base nesta mesma inspiração. Note-se que a penalização e a recompensa são o binómio central de todo o treino de animais, sendo pois um mecanismo de aprendizagem estabelecido e muito estudado, em áreas como Inteligência Artificial, mas pouco ou nada transposto para aplicações de hipertexto (, até ao STIGH).

Isto sugere que uma página com **cStighs** está a aprender as preferências dos seus utilizadores, por auto-domesticação.

### *10.3.4. Possibilidade no sentido de métricas mais refinadas, de utilização de websites*

A autonomia de cada hiperligação stigmérgica permite explorá-la em sentidos que não foram necessários para o STIGH. É muito simples registar, aquando de um click, mais informação do que o IP do cliente e um número de sessão. Estes dois valores correspondem, respectivamente, aos identificadores `REMOTE_ADDR` e `INSTANCE_ID`, do ambiente CGI (*Common Gateway Interface*).

O ambiente CGI é uma tecnologia muito madura, para a concepção de aplicações cliente-servidor, estando disponível em rigorosamente todos os servidores web. Esta tecnologia estabelece um espaço de nomes, ditos variáveis de ambiente. A estes nomes correspondem dados. Por exemplo, como já referido, ao nome `REMOTE_ADDR` corresponde o IP do cliente, e a `INSTANCE_ID` corresponde uma identificação numérica da sessão. Assim, em termos programáticos, não é necessário qualquer esforço para determinar o IP e o n° da sessão: basta saber consultar os identificadores certos, facultados pelo ambiente.

Há muitos outros nomes disponíveis. Por exemplo, no ambiente que serve o website [stigh.org](http://stigh.org), estão disponíveis 54 variáveis de ambiente CGI. Tipicamente, nem todas estarão valoradas (por exemplo, visitas directas a uma página, deixam em branco `HTTP_REFERER` – a página de proveniência). A mensagem é que podem recolher-se muitos mais dados directos, para além do IP e da sessão.

A generalidade das aplicações, para a recolha de informações da utilização de um website (*web metrics software*) regista todos estes valores, mas só até ao nível de detalhe da página. O que cada

**cStigh** poderia fazer de diferente, seria aumentar o detalhe dos registos, descendo a granularidade até à hiperligação e assim contribuir para uma nova geração de métricas, mais profundas.

## **11. Anexos**

**11.1. Versão de demonstração:** <http://stigh.org>

**11.2. Galeria de diagramas de fluxo:** <http://stigh.org/album/fd>

**11.3. Galeria de diagramas de sequência:** <http://stigh.org/album/sd>

**11.4. Código fonte:** <http://stigh.org/sc>

## Referências

- Antônio Dias de Figueiredo, P. R. C. 2006, 'Action Research and Design in Information Systems: Two Faces of a Single Coin', in *Information Systems Action Research: An Applied View of Emerging Concepts and Methods*, ed. N. Kock, Springer-Verlag, N.Y., pp. 61-96.
- Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G. & Bonabeau, E. 2003, *Self-Organization in Biological Systems (Princeton Studies in Complexity)*, Princeton University Press.
- Cameron Marlow, M. N., Danah Boyd, Marc Davis. 2006, 'Tagging Paper, Taxonomy, Flickr, Academic Article, To Read', in *ACM Hypertext 2006*, pp. 31-39.
- Creswell, J. W. 2002, *Research Design: Qualitative, Quantitative and Mixed Method Approaches*, Sage Publications Ltd.
- Czerner, T. 2002, *What Makes You Tick?: The Brain in Plain English*, Wiley.
- Fallman, D. 2003, 'Design-oriented Human-Computer Interaction', in *Conference on Human Factors in Computing Systems (CHI)*, pp. 225-232.
- Grassé, P. P. 1977, *Evolution of living organisms: Evidence for a new theory of transformation*, Academic Press.
- Humberto R. Maturana, F. J. V. 1992, *The Tree of Knowledge: Biological Roots of Human Understanding*, Shambhala Publications Inc., U.S.
- L. A. Carr<sup>1</sup>, D. D. R., W. Hall, G. Hill. 1998, 'Implementing an Open Link Service for the World-Wide Web', *World Wide Web Journal*, vol. 1, pp. 61-71.
- Laurel, B. 1993, *Computers as Theatre*, Addison Wesley.
- Leuf, B. 2006, 'Enhancing the Web', in *The Semantic Web - Crafting Infrastructure for Agency*, Wiley, pp. 3-30.
- Marais, E. N. 1969, *The soul of the white ant*, Kraus Reprint.
- Nelson, T. 1965, 'A File Structure for the Complex, the Changing, and the Indeterminate', in *ACM, 20th National Conference*, pp. 84-100.

- Nelson, T. 1970, 'No More Teachers' Dirty Looks', in *Computer Decisions*, pp. 16-23.
- Reeves, L. , *Insipid*, [Online], Available from: <<http://www.neuro-tech.net/insipid/>>.
- Rosenberg, J. 2000, 'Hypertext in the Open Air: A Systemless Approach to Spatial Hypertext', in *ACM Hypertext 2000*.
- Sala. , *websitesasgraphs*, [Online], Available from: <<http://www.aharef.info/static/htmlgraph/>>.
- Samhaa R. El-Beltagy, W. H., David De Roure, Leslie Carr. 2001, 'Linking in Context', in *ACM Hypertext 2001*, Aarhus, Denmark, pp. 151-160.
- Sousa, I. d. , *Por uma Internet auto-organizada como um formigueiro*, [Online].
- Vaishnavi, V. & Kuechler, B. (2005/04/26), *Design Research in Information Systems*, [Online], Available from: <<http://www.isworld.org/Researchdesign/drisISworld.htm>>.
- W3C. , *XML Linking Language (XLink)*, [Online], Available from: <<http://www.w3.org/TR/xlink/>>.
- Walker, J. 2005, 'Feral Hypertext: When Hypertext Literature Escapes Control', in *ACM - Hypertext 05*, Salzburg, Austria, pp. 46-53.
- Wardrip-Fruin, N. 2004, 'What Hypertext Is', in *ACM Hypertext 2004*, Santa Cruz, CA, USA, pp. 126-127.