



## 16º Congresso Nacional de Iniciação Científica

**TÍTULO:** ADOÇÃO DE QR-CODE COLORIDO COMO PROPOSTA PARA REPRESENTAÇÃO VISUAL E INTERAÇÃO EM GRANDES VOLUMES DE DADOS

**CATEGORIA:** CONCLUÍDO

**ÁREA:** CIÊNCIAS EXATAS E DA TERRA

**SUBÁREA:** COMPUTAÇÃO E INFORMÁTICA

**INSTITUIÇÃO:** INSTITUTO MUNICIPAL DE ENSINO SUPERIOR DE ASSIS

**AUTOR(ES):** GABRIEL YOSHIHARU RODRIGUES USSUY

**ORIENTADOR(ES):** GUILHERME DE CLEVA FARTO

Realização:



Apoio:



## RESUMO

De acordo com estatísticas apresentadas pela IBM (2015), diariamente são gerados aproximadamente 2,5 quintilhões de *bytes* de dados. Inclusive, tecnologias emergentes como, por exemplo, Internet das Coisas (*Internet of Things* – IoT) contribuem com a massiva geração de dados providos por sensores e outros dispositivos. Visando este aumento exponencial da quantidade de dados, esta pesquisa científica propõe a definição de uma nova abordagem para a expansão da capacidade de armazenamento em QR Codes. Além da investigação científica acerca da adoção de QR Codes e dos trabalhos relacionados, esta pesquisa objetiva a modelagem de uma abordagem classificada como “*Layer Colored QR Code*” que possibilita a geração, bem como a manipulação de QR Codes com maior capacidade de armazenamento. Para isso, a enfoque proposto se baseia no uso de QR Codes coloridos resultantes da unificação de múltiplas camadas (*layers*) de um conjunto de QR Codes tradicionais. Inicialmente, definiu-se uma metodologia sustentada pela revisão bibliográfica a partir dos trabalhos técnico-científicos que se relacionam ao contexto de QR Code customizado. Posteriormente, buscou-se contribuir por meio da modelagem (*design*) e implementação de uma API reutilizável e extensível em Java para avaliar, experimentalmente, a abordagem relatada. Como resultados, destaca-se que o uso de *Layer Colored QR Code*, de fato, auxilia na expansão da quantidade de informações em um QR Code. Também se ressalta que a avaliação experimental da abordagem evidenciou que o modelo dispõe de relevantes características como resistência a danos e sujeiras em até 30% dos defeitos verificados.

## PALAVRAS-CHAVE

QR Code; Java; Dados; Capacidade; Armazenamento; QR Code colorido; API

## 1. INTRODUÇÃO

A utilização do código de barras tornou-se amplamente popular em virtude de sua velocidade, precisão e características funcionais de leitura. Em consequência de sua popularização e aceitação mundial, o mercado passou a exigir códigos apropriados para armazenar mais informações, com maior variedade de caracteres e com menor

espaço de impressão. No entanto, os esforços realizados em aumentar a capacidade de armazenamento em códigos de barras resultaram no aumento da complexidade de leitura, bem como dos custos de impressão (GRILLO et al., 2010).

Devido às limitações dos códigos de barras, uma equipe de desenvolvimento da DENSO WAVE iniciou, em 1994, a implementação de um novo modelo bidimensional, anunciado como *Quick Response Code* (ou QR Code). Tal modelo de código de barras apresenta velocidade de leitura juntamente com maior capacidade de armazenamento em relação aos demais códigos de barra (QR CODE, 2015).

Desde o lançamento do QR Code, embora originalmente designado para catalogação de componentes da indústria automobilística, sua utilização tem sido generalizada, expandindo rapidamente às mais diversas áreas, tais como manufatura, varejo, saúde, biologia, transporte, *marketing* e demais (DENSO ADC, 2011).

Atualmente, um dos grandes desafios na área de computação se refere à manipulação e ao processamento de grandes quantidades de dados como no contexto de *Big Data* (VIEIRA et al., 2012). De acordo com dados da IBM (2015), todos os dias são gerados 2,5 quintilhões de *bytes* de dados, sendo que 90% dos dados no mundo foram obtidos por meio de sensores climáticos, redes sociais, fotos digitais, vídeos, registros de transações de compras e dispositivos móveis apenas nos últimos dois anos.

Visando uma real necessidade de novas abordagens e tecnologias para lidar com o extenso volume e complexidade de dados gerados, este trabalho propõe a modelagem e a implementação de um componente de software (API) para a geração e manipulação de QR Codes multicoloridos, focando-se em recursos de armazenamento e processamento de dados.

## **2. OBJETIVOS**

Este projeto pretende, como objetivos gerais, investigar os conceitos e recursos de códigos baseados no modelo QR Code, bem como sua aplicação prática por meio da plataforma Java. Pretende-se, como contribuição singular, relatar a modelagem (*design*) e a implementação de uma API, componente de software reutilizável e extensível, que possibilite maior capacidade de armazenamento quando comparado às características do modelo original. Para isso, esta pesquisa propõe a geração e a manipulação de QR Codes coloridos que unificam múltiplas camadas (*layers*) de um

conjunto válido de QR Codes tradicionais. Devido a estas propriedades, definiu-se o nome de “*Layer Colored QR Code*” à abordagem apresentada.

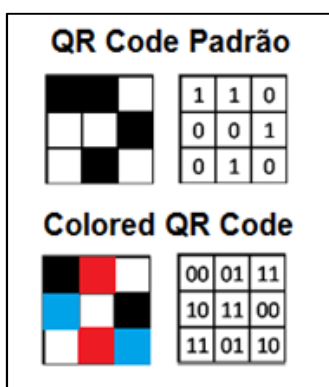
A condução deste trabalho pode ser orientada pelos seguintes objetivos específicos:

- (i) Pesquisar e explorar os conceitos e aspectos de QR Code;
- (ii) Modelar uma proposta baseada em QR Codes coloridos para a unificação e expansão da capacidade – definida como “*Layer Colored QR Code*”;
- (iii) Implementar uma API em Java a partir da especificação dos conceitos definidos, bem como dos resultados esperados;
- (iv) Experimentar e relatar os resultados obtidos na geração e na manipulação dos artefatos por meio da abordagem proposta.

Devido às vantagens de QR Code e à pouca quantidade de pesquisas que propõem e implementam tecnologias para o contexto de QR Code colorido, ressalta-se a importância deste trabalho. Por meio da estratégia apresentada e discutida, pretende-se disponibilizar uma versão refinada do QR Code tradicional, expandindo, dessa forma, sua capacidade de armazenamento.

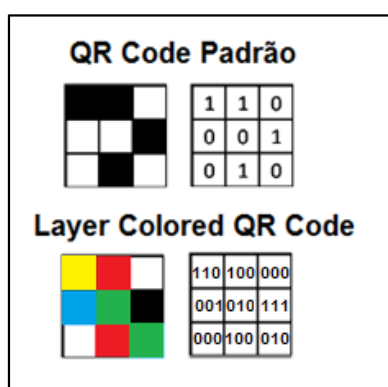
### 3. METODOLOGIA

Em um QR Code padrão, cada módulo representa um único *bit*, seguindo uma regra simples: o valor “1” para quadrados pretos e o valor “0” para quadrados brancos. Com a finalidade de se obter uma maior capacidade para o armazenamento de informações, propostas introduziram o conceito de cores na representação original. Entretanto, a maior parte das abordagens com QR Code coloridos utilizam apenas o



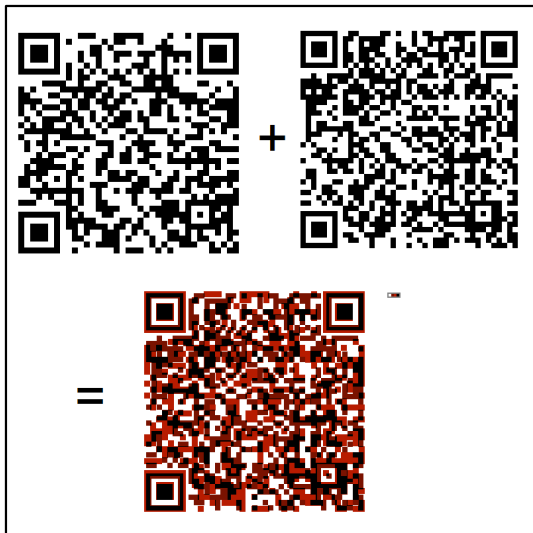
**Figura 1** – Módulos em *Colored QR Code*  
**Fonte:** Adaptado de GRILLO et al., 2010

Na abordagem proposta de *Layer Colored QR Code*, várias cores podem ser utilizadas, possibilitando assim a expansão da capacidade de armazenamento de dados. Portanto, quanto maior a quantidade de cores utilizadas para a geração do *Layer Colored QR Code*, maior será a capacidade de armazenamento de dados. Isto, pois, dessa forma, se permite a combinação de N QR Codes. Na Figura 2 é exemplificada a geração de um fragmento que compõe um QR Code colorido a partir da abordagem dos autores.



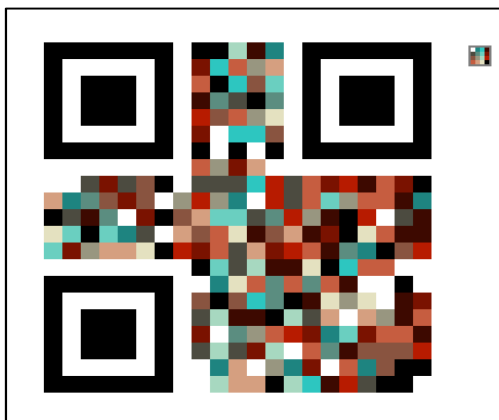
**Figura 2** – Módulos em *Layer Colored QR Code*  
**Fonte:** Adaptado de GRILLO et al., 2010

A geração do QR Code multicolorido em *Layer Colored QR Code* ocorre mediante a sobreposição de QR Codes padrões, aumentando a capacidade de armazenamento a cada sobreposição. A cada sobreposição realizada, também se aumenta o número de cores necessárias para representar as possíveis combinações binárias (0s e 1s) dos QR Codes que são utilizados processo de unificação. Para a identificação da quantidade de cores e sobreposições realizadas durante o processo de leitura e decodificação, esta abordagem também propõe o uso de um mapa de cores ao lado do QR Code colorido. Na Figura 3 é apresentado um QR colorido em sua totalidade, exemplificando a combinação de dois QR Codes tradicionais.



**Figura 3** – QR Code gerado em *Layer Colored Qr Code*  
**Fonte:** Elaborado pelos autores

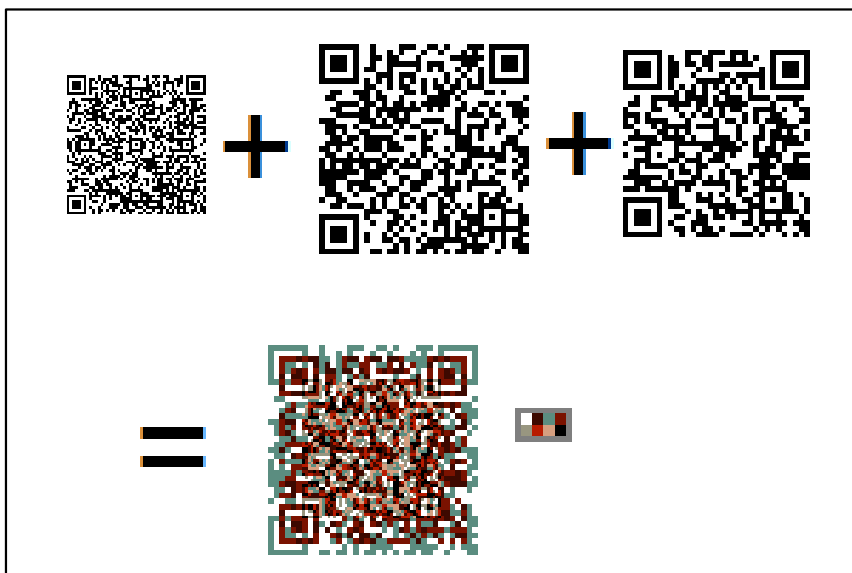
Para a identificação da versão do *Layer Colored QR Code*, juntamente com as cores utilizadas, um mapa de cores é definido para a interpretação QR Code colorido. Na geração, uma região específica é reservada para o mapa de cores (canto superior direito), possibilitando que N cores sejam interpretadas. Na Figura 4 é apresentado um mapa de cores ao lado do QR Code gerado com *Layer Colored QR Code*.



**Figura 4** – Exemplo de *Layer Colored QR Code*  
**Fonte:** Elaborado pelos autores

O mapa de cores é exibido em forma de matriz com três cores por linha e com tamanho de bloco configurável. Para a criação de um *Layer Colored QR Code*, com três *layers*, utiliza-se a seguinte sintaxe `[layer][layer]layer`. Seguindo a fórmula  $2^{\text{[quantidade de layers]}}$ , obtém-se oito cores. Na Figura 5 é apresentado um QR Code com três *layers*

(ou oito possíveis cores). Ressalta-se que, não necessariamente, os QR Codes utilizados devem estar com as mesmas dimensões ou até configurações.



**Figura 5** – QR Code três camadas (*layers*)  
**Fonte:** Elaborado pelos autores

#### 4. DESENVOLVIMENTO

Para a implementação da API da abordagem de *Layer Colored QR Code*, utilizou-se a biblioteca *ZXing* (“*Zebra Crossing*”), projeto *open source* de processamento de código de barras 2D.

Para a geração de um QR Code com múltiplas camadas, por intermédio da API proposta, deve-se instanciar um objeto do tipo *LayerColoredWriter*. (i) especificar os métodos de codificação das camadas, (ii) gerar a paleta de cores, (iii) informar os textos que serão utilizados para a geração das camadas, bem como (iv) os caminhos para a saída dos QR Codes padrões que compõem o QR Code colorido. Na Listagem 1 é apresentado um fragmento de software que adota a API de *Layer Colored QR Code* para geração.

**Listagem 1:** Fragmento de software com API da abordagem proposta (geração de QR Code colorido)  
**Fonte:** Elaborado pelos autores

```
public static void main(String[] args) throws IOException {  
  
    // Lista dados a serem utilizados na geração dos N QR Codes tradicionais  
    List<String> inputList = new ArrayList<>();  
}
```

```

inputList.add("Texto para o QR Code de número 1");
inputList.add("Texto para o QR Code de número 2");
inputList.add("Texto para o QR Code de número 3");

// Lista de arquivos de saída da geração de QR Codes
List<String> listPath = new ArrayList<>();
listPath.add("C:\\Users\\GYRU\\Desktop\\QR Codes\\QRCode1.png");
listPath.add("C:\\Users\\GYRU\\Desktop\\QR Codes\\QRCode2.png");
listPath.add("C:\\Users\\GYRU\\Desktop\\QR Codes\\QRCode3.png");

// Instanciação de API de Layer Colored QR Code
LayerColoredWriter lyr = new LayerColoredWriter();

// Configurações para QR Codes (codificação, nível de correção e margem)
lyr.setCharacterSet("UTF-8");
lyr.setErrorCorrection(ErrorCorrectionLevel.L);
lyr.setMargin(2);

// Método para geração de paleta dinâmica de cores (abordagem proposta)
lyr.generatePallete(3);

// Métodos para geração de QR Code colorido por meio da abordagem proposta
lyr.generateQRCode(inputList);
lyr.generateColoredQRCode(listPath, 400); }

```

Para a leitura dos QR Codes com múltiplas camadas, deve-se instanciar um objeto do tipo *LayerColoredReader*. (i) invocar o método para leitura da paleta de cores, (ii) informar o caminho de saída dos QR Codes e, por último, (iii) invocar o método que efetua leitura (transformação) do QR Code colorido em N QR Codes tradicionais. Na Listagem 2 é apresentado um fragmento de software que adota a API de *Layer Colored QR Code* para leitura.

**Listagem 2:** Fragmento de software com API da abordagem proposta (leitura de QR Code colorido)

**Fonte:** Elaborado pelos autores

```

public static void main(String[] args) throws IOException {

    // Instanciação de API de Layer Colored QR Code
    LayerColoredReader lyr = new LayerColoredReader();

    // Método para leitura de paleta de cores
    lyr.readPallete(
        "C:\\Users\\GYRU\\Desktop\\QR Codes\\ColoredQRCode.png");

    // Configuração de diretório de saída após transformação
    lyr.setOutputPath("C:\\Users\\GYRU\\Desktop\\QR Codes");

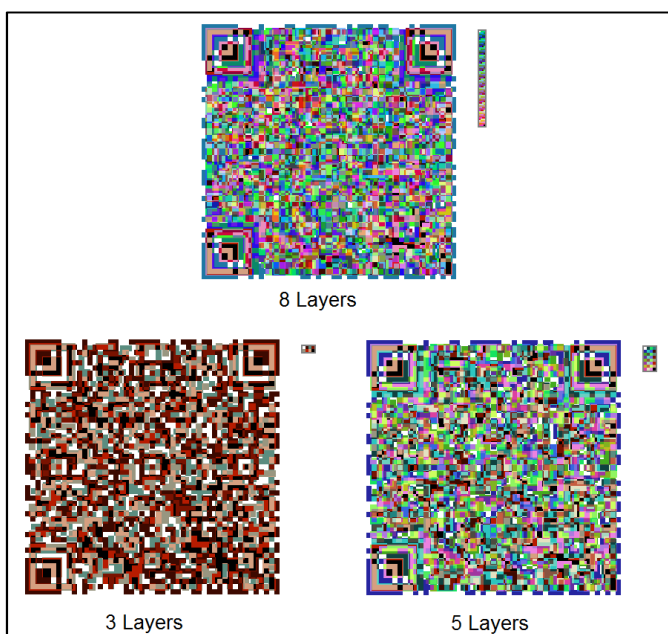
    // Método para extração de N QR Codes a partir da abordagem proposta
    lyr.readLayerColored(); }

```



## 5. RESULTADOS

Identificou-se que a abordagem proposta de *Layer Colored QR Code*, avaliada por meio da API desenvolvida em Java, fornecem evidências acerca da expansão da capacidade, podendo-se armazenar mais dados em relação ao QR Code tradicional. Por se tratar de uma sobreposição de vários QR Codes, a capacidade de armazenamento varia de acordo com a versão dos QR Codes sobrepostos, bem como a partir da quantidade de sobreposições realizadas. Quanto maior a quantidade de sobreposições realizadas, maior será a capacidade máxima de caracteres armazenados. Na Figura 6 são apresentados QR Codes com 3, 5 e 8 sobreposições, resultantes da geração a partir da abordagem e API propostas.

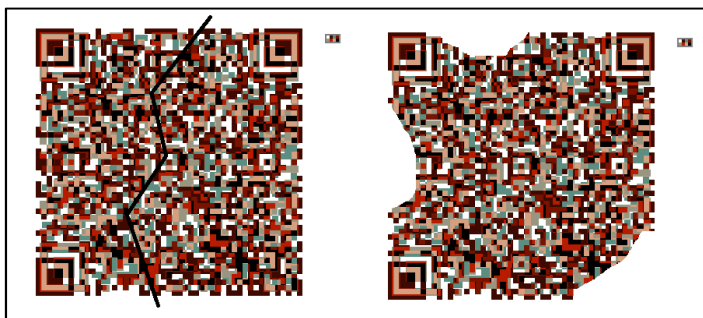


**Figura 6** – *Layer Colored QR Code* com 3, 5 e 8 camadas  
**Fonte:** Elaborado pelos autores

A cada sobreposição realizada, o número de cores necessárias para codificação cresce exponencialmente de acordo com a expressão  $2^N$ , onde  $N$  é o número de sobreposições. Assim, para a geração com 23 sobreposições, são necessárias 8.388.608 combinações de cores ( $2^{23}$ ). Cada cor do RGB (*red*, *green* e *blue*) varia de 0 a 255, sendo possível 16.581.375 combinações de cores ( $255 \times 255 \times 255$ ). Dessa forma, o número máximo de sobreposições possíveis, levando em consideração o padrão RGB é de 23 QR Codes. Isto, pois a sobreposição de 24 QR Codes exige uma combinação de 16.777.216 cores. Ressalta-se que, embora sejam possíveis 23

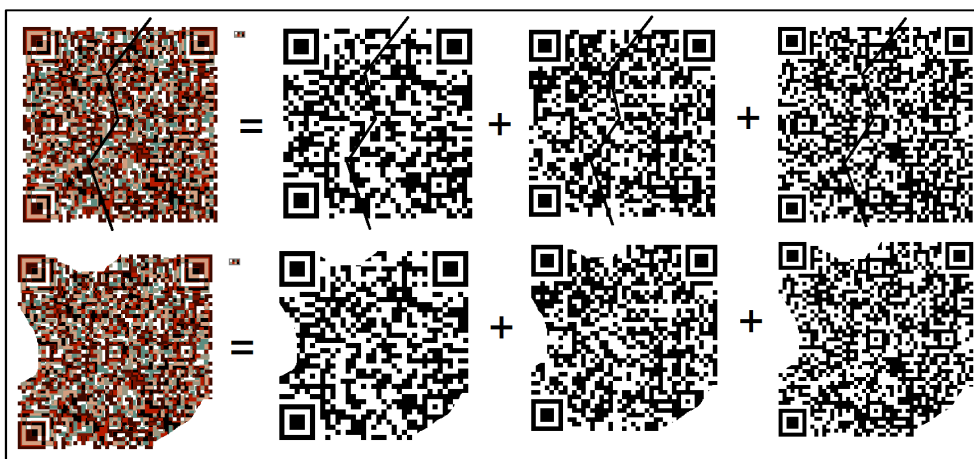
sobreposições, o dispositivo de reconhecimento do QR Code deve ser capaz de identificar todas as combinações utilizadas, caso contrário a codificação torna-se inútil.

O QR Code gerado a partir da API *Layer Colored QR Code* manteve a característica de resistência a dano e sujeira, apresentados no QR Code padrão. Verificou-se que foi possível restaurar de 7 a 30% da imagem danificada, de acordo com o nível de correção de erros utilizado. Na Figura 7 são apresentados exemplos de QR Codes danificados que puderam ser restaurados na leitura do código.



**Figura 7** – *Layer Colored QR Codes* danificados  
**Fonte:** Elaborado pelos autores

Durante a avaliação experimental, identificou-se danos são replicados aos demais QR Codes gerados no processo de leitura, porém estes ainda são considerados válidos para a extração dos dados. Destaca-se que, apesar de interpretar QR Codes avariados, danos em áreas críticas como a região de detecção de posição, podem inviabilizar a leitura de um ou mais QR Codes sobrepostos. Na Figura 8 são apresentados QR Codes coloridos, porém danificados, bem como seu processo de leitura ou transformação em QR Codes tradicionais.



**Figura 8** – Danos do Colored QR Code replicados aos QR Codes resultantes  
**Fonte:** Elaborado pelos autores

## 6. CONSIDERAÇÕES FINAIS

Por meio da abordagem modelagem e da API implementada, torna-se possível armazenar uma maior quantidade de dados em um QR Code a partir da sobreposição de vários QR Codes como proposto por este trabalho.

Apesar de a API proposta ter sido experimentada e aplicada apenas em um contexto experimental, espera-se validá-la em cenários e ambientes reais. Assim, os resultados obtidos podem contribuir com melhorias ao conceito original de QR Code.

Os QR Codes gerados a partir da API puderam ser utilizados para expandir a capacidade de armazenamento de dados. A biblioteca implementada em Java expõe funcionalidades para a geração e a leitura de QR Codes coloridos que podem ser aplicados a diferentes cenários. Além disso, mantiveram as características de resistência a dano e sujeira, também apresentados no QR Code tradicional. Mesmo com ruídos (defeitos) no QR Code colorido, a leitura foi válida e esta resulta na transformação em N QR Codes que originalmente foram utilizados na sobreposição.

Em trabalhos futuros, a abordagem proposta pode ser adaptada e evoluída para distintos contextos como saúde e educação. Particularidades identificadas na adoção real de *Layer Colored QR Code* podem orientar novas funcionalidades e recursos. Também se sugere a integração da API desenvolvida em dispositivos móveis como, por exemplo, as plataformas Android, iOS da Apple e Windows Phone da Microsoft.

## REFERÊNCIAS

DENSO ADC. **QR Code Essentials**. Disponível em: <<http://denso-adc.com/download>>, 2011. Acessado em 11 de Julho de 2016.

GRILLO, A.; LENTINI, A.; QUERINI, M.; ITALIANO, G. F. **High Capacity Colored Two Dimensional Codes**. In: Proceedings of the International Multiconference on Computer Science and Information Technology, 709 p., 2010.

IBM. **What is Big Data?** Disponível em: <<http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html>>, 2015. Acessado em 20 de Julho de 2016.

QR CODE. **History of QR Code**. Disponível em <<http://www.qrcode.com/en/history/>>. Acessado em 15 de Agosto de 2016.

VIEIRA, M. R.; FIGUEIREDO J. M.; LIBERATTI, G.; VIEBRANTZ, A. F. M. **Bancos de Dados NoSQL: Conceitos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data**. Minicurso apresentado no Simpósio Brasileiro de Bancos de Dados, 2012.