



INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DE PERNAMBUCO

Campus Recife

Coordenação de Sistemas de Informação

DEPARTAMENTO ACADÊMICO DE SISTEMAS, PROCESSOS E CONTROLES
ELETRO-ELETRÔNICOS - DASE

EFRAIM DE SOUZA RAMOS FILHO

**APLICAÇÃO DE UM MÓDULO DE CONSULTA DE INFORMAÇÕES EM UMA
BIBLIOTECA COMUNITÁRIA: A Open QRBib**

RECIFE
2018

EFRAIM DE SOUZA RAMOS FILHO

**APLICAÇÃO DE UM MÓDULO DE CONSULTA DE INFORMAÇÕES EM UMA
BIBLIOTECA COMUNITÁRIA: A OPEN QRBIB**

Trabalho de Conclusão de Curso
Apresentado ao curso de Tecnologia em
Análise e Desenvolvimento de Sistemas,
como pré-requisito para obtenção do título de
Tecnólogo em Análise e Desenvolvimento de
Sistemas pelo Instituto Federal de Educação,
Ciência e Tecnologia de Pernambuco - IFPE.

Orientador: Paulo Abadie Guedes.

Ficha elaborada pela bibliotecária Ana Lia Moura CRB4/974

R175a

Ramos Filho, Efraim de Souza.

Aplicação de um módulo de consulta de informações em uma biblioteca comunitária:
a open QRbib / Efraim de Souza Ramos Filho. Recife, PE: O autor, 2018.

40f. il. color. : 30 cm.

TCC (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal de
Pernambuco, DASE 2018.

Inclui Referências.

AGRADECIMENTOS

A minha mãe e aos meus amigos, que fizeram com que essa jornada fosse muito melhor do que ela deveria ser.

“Cada um terá a vista da montanha que subir”.
ÍCARO FONSECA

RESUMO

O projeto Open QRBib consiste em um sistema para agilizar a busca por informações de livros em bibliotecas comunitárias, principalmente em livros que estão danificados, por meio do uso de QR Codes. Estes códigos de barras bidimensionais são capazes de armazenar blocos de textos interativos que podem ser lidos como múltiplos tipos de textos diferentes, como por exemplo, URLs, números de telefone, e-mails, localizações georreferenciadas, entre outros formatos. O sistema será utilizado para transformar informações a respeito de livros de bibliotecas comunitárias em QR Codes, pois estas bibliotecas raramente possuem um sistema de terminais apropriado para consulta de tal conteúdo. Os dados gerados ao final da execução do projeto serão impressos e fixados nos livros para que a consulta possa ser realizada pela pessoa interessada em determinado livro, cada livro deverá possuir dois QR Codes fixados em sua capa, onde um deverá conter informações mais básicas a respeito do livro e o outro será um *link* que levará o usuário a uma wiki onde maiores informações do livro deverão ser mostradas.

Palavras-chave: QR Code. Livros. Bibliotecas Comunitárias.

ABSTRACT

The Open QRBib project contains a system to speed up a search for information from books in community libraries, especially damaged books, through the use of QR Codes. These two-dimensional barcodes are capable of storing blocks of interactive texts that can be read as multiple types of different texts, such as URLs, phone numbers, emails, georeferenced locations, among other formats. The System will be used to transform information regarding books from community libraries into QR Codes, because these libraries rarely have an appropriate terminal system for querying such content. The data generated at the end of the project execution will be printed and fixed in the books so the query can be performed by the person interested in a particular book, each book shall have two QR Codes fixed on its cover, where one should contain basic information about the book, and the other will be a link that will take the user to a wiki page, where more information from the book should be shown.

Keywords: QR Code. Books. Community Libraries.

LISTA DE ILUSTRAÇÕES

Figura 1 - Livro danificado, “O Exorcista” nas visões de verso, lateral e frente.....	14
Figura 2 - Um QR Code	16
Figura 3 - Capacidade de Armazenamento.....	20
Figura 4 - Redução do tamanho.....	20
Figura 5 - Codificando caracteres japoneses.	21
Figura 6 - Resistência à danos e sujeira.	22
Figura 7 - Padrões de detecção de posicionamento.	23
Figura 8 - Acréscimo estruturado de informações.....	23
Figura 9 - Diagrama de caso de uso.	27
Figura 10 - Escopo do projeto.	28
Figura 11 - Disposição dos QR Codes no documento PDF.	29
Figura 12 - Método de configuração de lista de livros.	33
Figura 13 - Método de geração de PDF	34
Figura 14 - Método de geração dos QR Codes.....	35
Figura 15 - Tela inicial do sistema.....	35
Figura 16 - Tela de inserção de livros.	36
Figura 17 - Tela de consulta de livros.	37
Figura 18 - Resultado da consulta.....	37

LISTA DE ABREVIATURAS E SIGLAS

QR Code	Quick Response Code
ISBN	International Standard Book Number
2D	Duas Dimensões
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
AIM	Identificação Automática de Fabricantes
JEIDA	Japan Electronic Industry Development Association
JIS	Japanese Industrial Standards
ITSC	Information Technology Standards Committee
PC	Personal Computer
JVM	Java Virtual Machine
IDE	Integrated Development Environment
JAR	Java ARchive
ECI	Extended Channel Interpretation
FNC1	Function Code One
API	Application Programming Interface
PDF	Portable Document Format
URL	Uniform Resource Locator

SUMÁRIO

1 INTRODUÇÃO	12
2 OBJETIVO GERAL	14
2.1 Objetivos Específicos	14
2.2 Estrutura do Trabalho	15
3 FUNDAMENTAÇÃO TEÓRICA	15
3.2 Funcionalidade do QR Code	16
3.2.1 Codificação	17
<i>3.2.1.1 Análise dos dados</i>	17
<i>3.2.1.2 Codificação dos dados</i>	17
<i>3.2.1.3 Codificação da correção de erros</i>	18
<i>3.2.1.4 Estruturação da mensagem final</i>	18
<i>3.2.1.5 Mascaramento</i>	19
3.3 Vantagens do QR Code	19
3.4 Aplicações do QR Code	23
4 O PROJETO OPEN QRBIB	25
4.1 Visão Geral	25
4.2 Descrição do software do sistema	25
4.3 Principais classes na implementação do sistema	29
4.4 Projeto e desenvolvimento do sistema	30
<i>4.4.1 Ambiente de Desenvolvimento</i>	31
4.5 O estudo das bibliotecas de geração de QR Code	32
4.6 Prova de conceito do sistema	33
5 CONSIDERAÇÕES FINAIS	38
REFERÊNCIAS	40

1 INTRODUÇÃO

Os investimentos globais em Tecnologia da informação estão crescendo a cada ano - segundo a empresa de consultoria Gartner, Inc. a previsão até o fim de 2018 deve chegar a aproximadamente 3.7 trilhões de dólares. Caracterizando um aumento de 6.2% em relação ao ano anterior - esses investimentos visam principalmente a maximização de recursos, bem como a facilitação de certos procedimentos. Dessa forma as tecnologias existentes então em constante adaptação e evolução.

Tendo em vista esse crescimento o código de barras certamente não poderia ficar de fora desse processo, o resultado de tal evolução foi o, já tão conhecido, *QR Code* (do inglês *Quick Response Code*). Diferentemente do seu antecessor, o código de barras unidimensional, que armazena apenas informações numéricas, o *QR Code* é capaz de armazenar informações alfanuméricas em suas duas dimensões, armazenando, dessa forma, uma quantidade de informações muito maior, se comparado com o código de barras tradicional.

O *QR Code* vem sendo amplamente utilizado em países mais desenvolvidos como o Japão, Estados Unidos, Reino Unido e Canadá, em países como o Brasil seu uso vem crescendo cada vez mais ao longo dos anos. O *QR code* pode ser aplicado nas mais variadas áreas, podendo ser encontrados em lojas, prédios supermercados e eventos, além de materiais publicitários de corporações de grande e médio porte, e algumas vezes até nas de pequeno porte. Para cada estabelecimento ou ocasião uma utilização diferenciada pode surgir, porém todos se utilizam de câmeras de aparelhos móveis como dispositivo de leitura e decodificação.

Desde sua criação o *QR Code* vem se adaptando às propostas do mercado globalizado e contribuindo para o aumento da interação entre clientes e organizações, sendo, em muitas ocasiões, um diferencial de marketing. Diante deste contexto, este projeto se propõe a exibir um outro funcionamento além dos já citados previamente, aplicando o *QR Code* em uma biblioteca comunitária, com o objetivo de agilizar as buscas e desafogar os voluntários que prestam serviço na

mesma, assim como mostrar de uma forma rápida e objetiva o conteúdo de cada livro para quem quiser consultá-los.

A tecnologia, normalmente, está diretamente ligada à comodidade, ou facilidade que certas ferramentas podem nos oferecer, seja diminuindo o tempo de execução de certas ações ou diminuindo o esforço aplicado para executar tais ações, e nestes pontos o QR Code pode ser aplicado, como agente facilitador no armazenamento e exibição de informações.

Com a criação da Open QR Bib será possível diminuir em muito o tempo de consulta de informações relativas à livros em bibliotecas comunitárias de médio e pequeno porte, de forma que os interessados em locar livros consigam de forma rápida e simples as informações necessárias a respeito de cada livro, sem que seja necessária a consulta verbal aos voluntários ou trabalhadores da biblioteca.

O sistema consiste em utilizar o conceito de armazenamento e exibição do QR Code no ambiente dessas livrarias. O objetivo primário do projeto é, uma vez feita a catalogação dos livros, tais dados serão armazenadas em uma base de dados, e a partir dela serão gerados os QR Codes contendo tais informações.

Esses dados também poderão ser alterados dinamicamente em por meio do TiddlyWiki, site que será alimentado pelos funcionários da biblioteca, onde as informações também deverão estar disponíveis para consulta do usuário.

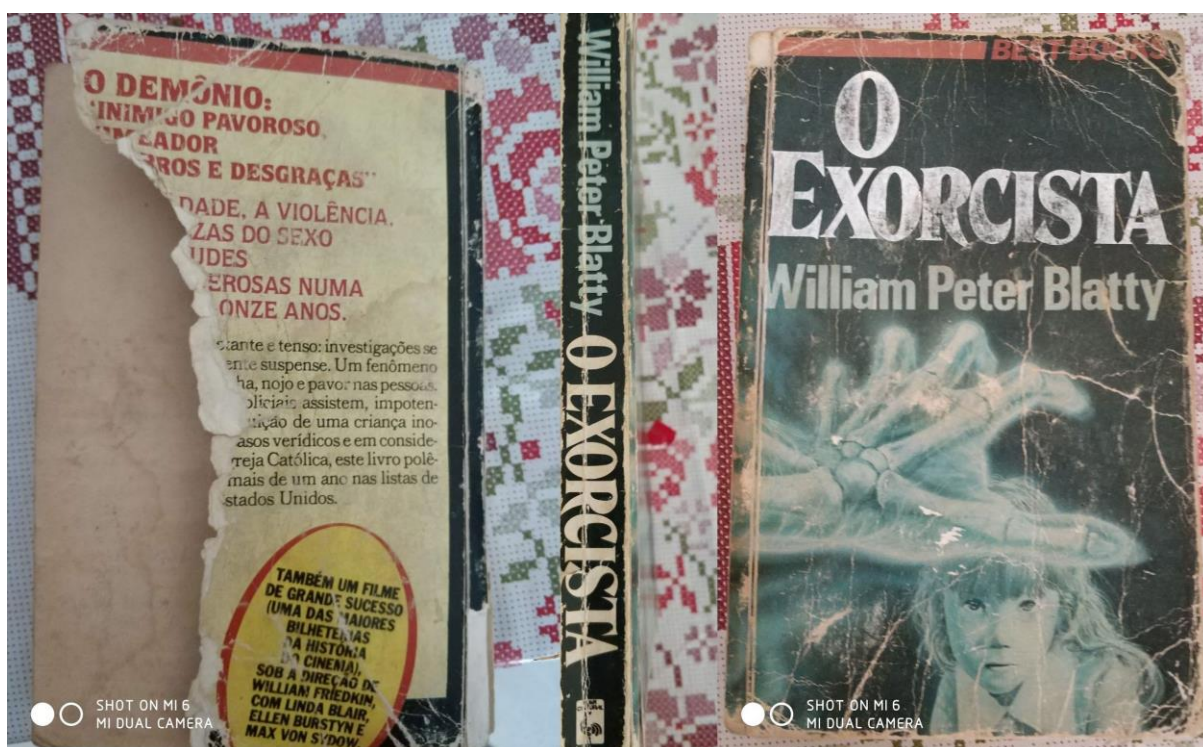
As informações contidas no QR Code poderão variar de biblioteca para biblioteca, porém as mais básicas deverão permanecer, são elas:

- Título da obra;
- Nome(s) do(s) autor(es);
- *Link* para o TiddlyWiki;
- ISBN (International Standard Book Number)
- Sinopse.

2 OBJETIVO GERAL

O objetivo geral do sistema é, oferecer um método de consulta das informações de determinados livros, em bibliotecas comunitárias, através do uso do QR Code, visando assim, agilizar as buscas por informações a respeito dos livros e desafogar os voluntários que prestam serviço em tais bibliotecas. Assim como a possível identificação de informações que foram perdidas, devido à danos, seja na capa ou contracapa, que dificultem ou impossibilitem a obtenção de tais informações dos livros, como mostra a Figura 1.

Figura 1 - Livro danificado, “O Exorcista” nas visões de verso, lateral e frente.



Fonte: Ramos, Ramos, Filho (2018)

2.1 Objetivos Específicos

Os Objetivos específicos deste trabalho são os seguintes:

- Estudar e analisar a biblioteca ZXing, responsável pela construção dos QR Codes.

- Projetar e implementar um serviço que armazene e gere os códigos quando necessários, para a fixação nos livros.
- Ajudar com a organização e consulta das bibliotecas comunitárias, uma vez que, com uma rápida escaneada no QR Code será possível saber em que seção determinado livro deverá ficar.
- Ajudar os leitores a identificar informações importantes, principalmente em livros com danificados de alguma forma, assim como os próprios livros, em casos mais extremos.

2.2 Estrutura do Trabalho

Este trabalho está estruturado da seguinte forma, no capítulo 2, é apresentada uma explicação da tecnologia estudada e utilizada na produção do trabalho. No capítulo 3, será descrito o funcionamento do sistema, assim como, o processo seu de desenvolvimento e sua estrutura, além disso, será apresentada uma breve explanação das principais classes e métodos que compõem o projeto como um todo. No capítulo 4 são apresentadas as devidas conclusões para o projeto, alguns pontos que dificultaram a conclusão do mesmo e o planejamento para trabalhos futuros.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 História do QR Code

Criado em 1994 o Qr Code é um código de barras de duas dimensões (2D) que foi desenvolvido no Japão, pela Denso Wave Corporation (uma subsidiária do grupo Toyota), tendo como objetivo ser um código de leitura fácil para o leitor, inicialmente usado no indústria automotiva para identificar as partes dos veículos durante o processo de manufatura.

Três anos após sua criação o QR Code foi aceito como um padrão internacional de AIM (Identificação Automática de Fabricantes). Em 1998, ele foi aprovado como padrão pela Associação para o Desenvolvimento da Indústria Eletrônica Japonesa, o JEIDA (do inglês *Japan Electronic Industry Development*

Association), sob o código JEIDA-55. Em 1999, aprovado como um padrão industrial japonês JIS (do inglês *Japanese Industrial Standards*) JIS X 510.

Atualmente, essa tecnologia é padronizada pela *International Organization for Standardization* sob o código ISO/IEC (*International Electrotechnical Commission*) 18004, e a patente está registrada pela Denso Wave Corporation, porém não é praticada pela empresa.

Diferentemente do seu antecessor, o código de barras tradicional, o QR Code é capaz de armazenar dados na vertical e horizontal.

Figura 2 - Um QR Code



Fonte: Rouse, Margaret (2013). Disponível em: <https://whatis.techtarget.com/definition/QR-code-quick-response-code>.

3.2 Funcionalidade do QR Code

Um símbolo QR Code é capaz de armazenar mais dados em determinada área do que um código de barras tradicional. Para se adequar a diferentes quantidades de dados, podem ser gerados QR Codes de tamanhos diferentes, chamados de versões, estas versões vão de 1 a 40, Denso Wave Incorporated (20??). Além de uma versão, cada símbolo tem um formato, que é o nível de correção de erros daquele QR Code em particular. Além de sua acomodação em

tamanho e falhas também é possível alterar sua coloração e design, para uma melhor acomodação em anúncios e afins.

3.2.1 Codificação

A transformação de um dado de entrada para uma saída em QR Code é chamada de codificação, esta codificação normalmente seguirá as etapas padrão, descritas no documento divulgado pela ISO/IEC em 2015, tal padrão está descrito nos tópicos a seguir:

3.2.1.1 Análise dos dados

Durante essa análise ocorre a identificação de que tipo de dados estará no QR Code, se será necessário alternar entre os diferentes modos, além de selecionar o nível de detecção e correção de erros que será aplicado. O programa escolherá a versão mais adequada de acordo com a quantidade de dados a serem codificados.

3.2.1.2 Codificação dos dados

De acordo com codificação dos dados deverá ocorrer da seguinte maneira: os dados serão convertidos em uma sequência de bits, de acordo com as regras definidas para cada modo, em seguida é inserido indicadores de modo antes da sequência de bits (caso haja mudança de modo, serão inseridos outros indicadores antes das próximas sequências), além de um terminador no final das sequências de dados. A sequência final de bits é composta por: indicador de modo + contador de caracteres + dados.

Cada tipo de dado possui um indicador diferente, são eles 0111 que representam o ECI (do inglês *Extended Channel Interpretation*), padrão técnico que define e especifica os métodos de sequência de escape ECI para transmitir informações de canal estendido, 0001 que representa os dados numéricos, 0010 que representa os dados alfanuméricos, 0100 que representa as sequências de 8 bits, 0011 que representa os anexos estruturados, 0101 e 1001 para os tipos FNC1, Código de Função Um (do inglês *Function Code One*), usado para dados formatados de acordo com especificações predefinidas (data de validade, número de lote) e por último o 0000 que é o terminador.

3.2.1.3 Codificação da correção de erros

“Dependendo da versão e nível de correção de erros, a sequência de codewords de dados será subdividida em um ou mais blocos” (ISO, 2015). Para cada um dos blocos será aplicado um algoritmo que gera os respectivos codewords de correção (o algoritmo de correção também será aplicado separadamente a cada um). Os codewords de correção de erros são colocados no final da sequência de dados

A Informação de formato deverá mostrar o nível de correção de erros empregado, existem 4 níveis distintos cada um deles possui a capacidade de corrigir uma porcentagem dos dados. Os níveis de correção são mostrados na Tabela 1:

Tabela 1 - Níveis de correção de erros

Nível de correção de erro	Capacidade de recuperação de dados em % (aproximadamente)
L	7
M	15
Q	25
H	30

Fonte: International Organization for Standardization (ISO). ISO/IEC 18004:2015. Information technology -- Automatic identification and data capture techniques -- QR Code bar code symbology specification. p. 36.

3.2.1.4 Estruturação da mensagem final

Os codewords de dados e de correção de erros são intercalados, adicionando bits remanescentes se necessário. Os codewords são geralmente no tamanho 4x2, às vezes sofrendo distorções. Os codewords de dados devem ser colocados primeiro na disposição, e, em seguida, os de correção. Eles são colocados começando do canto inferior direito, de baixo para cima. Quando chegam

à extremidade superior, os dados passam a ser colocados de cima para baixo na próxima coluna à esquerda, e assim alternadamente.

3.2.1.5 Mascaramento

Existem 7 padrões de mascaramento diferentes, cada um deles com um número binário de referência e uma condição que gera o padrão de bits do modelo. Cada padrão é aplicado à região de código do símbolo, através da operação XOR, e é escolhido aquele que otimiza o balanceamento entre pretos e brancos e minimiza a ocorrência de padrões indesejáveis. Por exemplo, não é desejável que na região de dados se encontre uma sequência de pretos e brancos semelhante à encontrada no padrão de detecção de posição.

3.3 Vantagens do QR Code

O QR Code possui múltiplas vantagens sobre seu antecessor, o código de barras comum, essas vantagens fizeram com que ele conseguisse se popularizar com velocidade em múltiplas áreas, as principais vantagens desta tecnologia são:

Alta capacidade de codificação de dados - Enquanto que o código de barras tradicional é capaz de armazenar aproximadamente 20 dígitos, o QR Code é capaz de armazenar uma quantidade dezenas de vezes maior, sendo capaz de armazenar dados numéricos, alfabéticos, kanjis, Katakana, Hiragana, símbolos, códigos binários e códigos de controle. aproximadamente 7.089 caracteres podem ser codificados em um único QR Code (DENSO [20_ _]).

Figura 3 - Capacidade de Armazenamento.



Fonte: QRcode.com (2018). Disponível em: <http://www.qrcode.com/en/about/>

Tamanho de impressão reduzido - Sendo capaz de armazenar informações tanto horizontalmente quanto verticalmente o QR Code é capaz de armazenar a mesma quantidade de dados em, aproximadamente, um décimo do espaço que um código de barras tradicional seria capaz, como mostrado na figura 4 (para tamanhos ainda menores existe o micro QR Code (DENSO [20_ _])).

Figura 4 - Redução do tamanho.

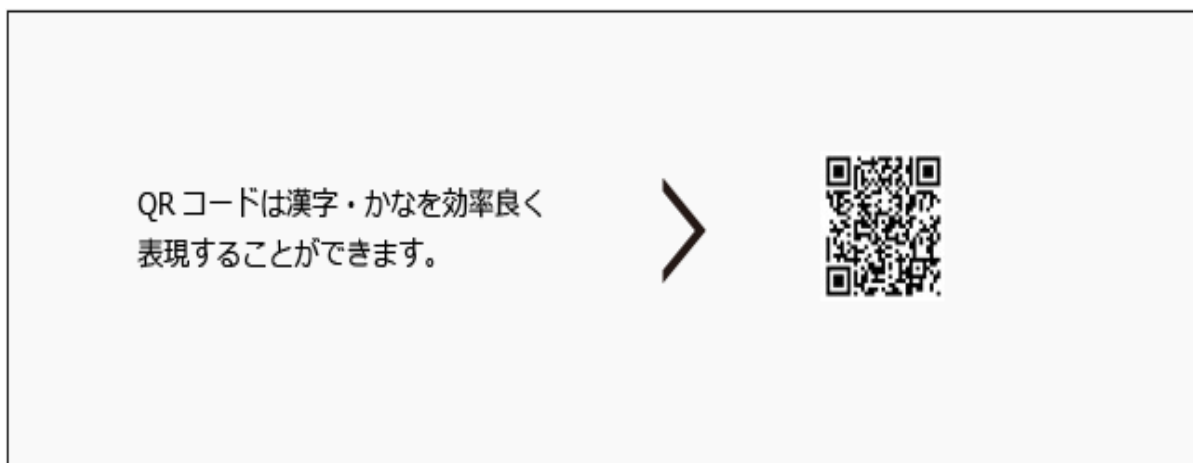


Fonte: QRcode.com (2018). Disponível em: <http://www.qrcode.com/en/about/>

Capacidade de armazenamento de Kanjis e Katakana - Como foi desenvolvido no Japão é natural que o QR Code seja capaz de codificar caracteres comuns no país, como Kanjis e Katanas. Um caractere desse tipo, em seu tamanho total é codificado em 13 bits, permitindo que o QR Code seja capaz de

armazenar 20% mais dados, se comparado a outras simbologias 2D (DENSO [20_ _]). .

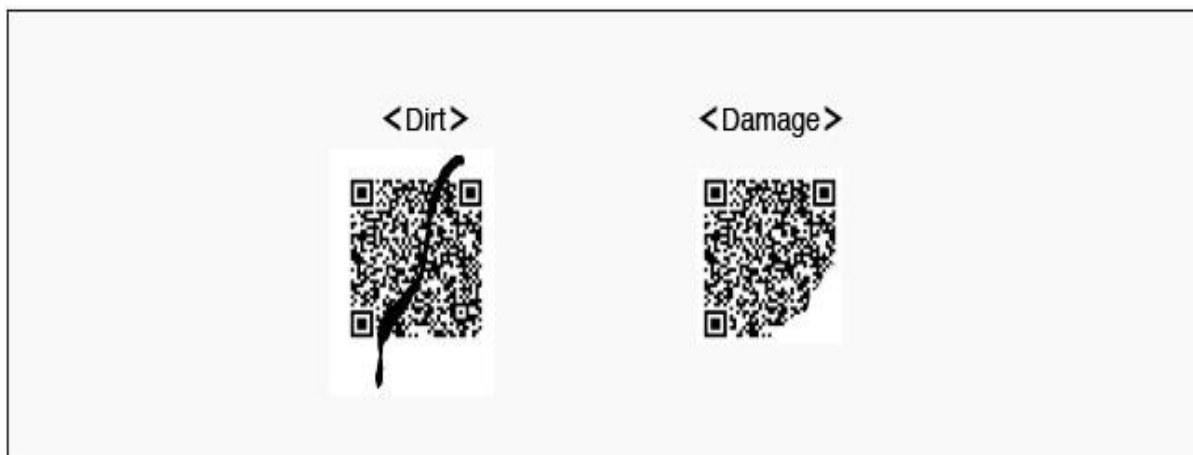
Figura 5 - Codificando caracteres japoneses.



Fonte: QRcode.com (2018). Disponível em: <http://www.qrcode.com/en/about/>

Resistência a sujeira e danos - O QR Code é capaz de detectar e corrigir erros, por meio de uma série de *codewords* (palavras de código), uma *codeword* é uma unidade que ajuda a construir as áreas de dados, no caso do QR Code uma *codeword* é igual a 8 bits e cada bit é representado por uma unidade (preto para 1 e branco para 0 (INTERNATIONAL (2015)). Graças as *codewords* é possível que os dados sejam processados e lidos mesmo que o QR Code esteja parcialmente sujo ou danificado, em alguns casos, a recuperação de dados pode não ser executada, pois a mesma depende do quanto o símbolo QR foi danificado ou do quão sujo ele está (DENSO [20_ _]). É possível recuperar os dados até uma margem de 30% de dano, dependendo do nível de correção de erros, como mostrado na Tabela 1.

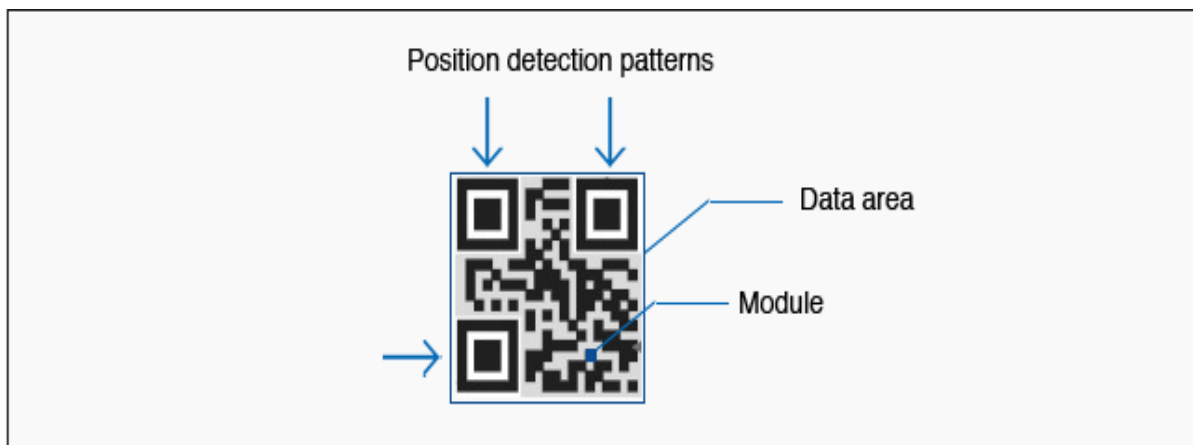
Figura 6 - Resistência à danos e sujeira.



Fonte: QRcode.com (2018). Disponível em: <http://www.qrcode.com/en/about/>

Capacidade de leitura qualquer direção (360°) - É possível realizar a leitura, em alta velocidade, de um QR Code em 360° de rotação. Isso é possível graças aos padrões de detecção de posicionamento localizados em três dos quatros cantos do símbolo. Esses padrões de posicionamento permitem uma leitura rápida e precisa independente do ângulo de rotação do símbolo, dessa forma é possível contornar possíveis interferências referente a angulação e do fundo DENSO [20_ _].

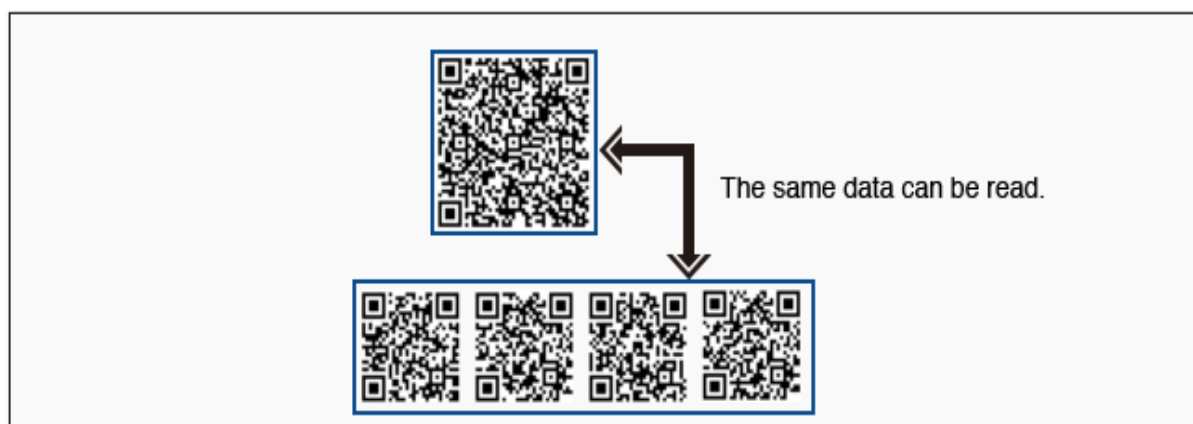
Figura 7 - Padrões de detecção de posicionamento.



Fonte: QRcode.com (2018). Disponível em: <http://www.qrcode.com/en/about/>

Acréscimo estruturado - O QR Code pode ser dividido em múltiplas áreas dentro do seu código de dados, de forma que ele é estruturado para acréscimo de informações, ou seja, as informações armazenadas em vários símbolos podem ser reconstruídas em um único símbolo, sem que ocorra perda de dados. Um símbolo QR pode ser dividido em até 16 sub símbolos, permitindo assim a impressão de várias informações independentes em uma área restrita (DENSO [20_ _]).

Figura 8 - Acréscimo estruturado de informações.



Fonte: QRcode.com (2018). Disponível em: <http://www.qrcode.com/en/about/>

3.4 Aplicações do QR Code

Ainda que tenha sido criado com propósito de identificar partes de veículos, o QR Code vem apresentando muitas outras funções ao longo dos anos, de modo

que a aplicação dessa tecnologia vem sendo ampliada cada vez mais ao com o passar do tempo. Aplicações de QR Code podem ser encontradas nas mais diferentes áreas de atuação, como por exemplo:

Identificação - O propósito inicial do QR Code foi a identificação de peças de veículos, armazenando identificadores primários, de forma que uma leitura rápida em um QR Code de uma determinada peça poderá distingui-la das demais.

Publicidade - O QR Code vem se adaptando cada vez mais ao mundo globalizado, contribuindo significativamente para o aumento da interação entre clientes e organizações, seja contribuindo para que as empresas a “se mostrem” mais ao público, ou até mesmo ajudando os clientes a conhecer mais sobre o trabalho e campo de atuação dessas empresas.

Segundo o Bôas (2008), em sua matéria para o jornal A Tarde da Bahia, a primeira empresa que utilizou o QR Code, no Brasil, para campanhas publicitárias foi a Fast Shop, loja de varejos e produtos eletrônicos, no ano de 2007. O código da campanha possuía um *link* para a página da loja, onde haviam ofertas e produtos.

Segurança Alimentar - Em grandes supermercados alguns produtos alimentícios possuem QR Codes impressos em suas embalagens, como afirma o professor do Departamento de Informática do Centro Técnico Científico da Pontifícia Universidade Católica do Rio de Janeiro (CTC/PUC-Rio) Raposo (2014), em sua entrevista para o Globo Ciência, tais códigos mostram a procedência do produto aos consumidores, assim como informações nutricionais adicionais, criando assim, uma maior confiança entre consumidor e produtor.

Turismo - Em países como o Japão, diversos museus e locais históricos utilizam o QR Code em suas ruas e monumentos, oferecendo informações históricas relevantes, relativas aos mesmos. No Brasil, o QR Code também é utilizado nas mais diversas áreas do turismo, um dos usos mais recentes desta tecnologia no país é em cemitérios, como de Santo Amaro na cidade de Recife - PE, que identifica túmulos de personalidades que marcaram a história do país, como mostrado no trabalho de Ramos; Chagas; Santos (2018). Além disso, companhias aéreas têm utilizado, desde 2010, esses códigos em seus bilhetes aéreos.

Saúde – De acordo com Soon (2008), em seu artigo para o ITSC (*Information Technology Standards Committee*), Hospitais do Japão e Hong Kong têm utilizado o QR Code, seja para a identificação dos pacientes e seus respectivos quadros, como também para o acompanhamento e identificação de visitantes. No caso dos pacientes o código é impresso em uma pulseira identificadora, onde normalmente, informações como nome, número de identificação e tratamento são exibidas, isso garante que o paciente correto, receberá o medicamento correto e o tratamento adequado, na hora certa, diminuindo assim os riscos de erro. Em outras localidades os médicos utilizam tablets para realizar a leitura dos códigos, fixado na entrada dos quartos dos pacientes, onde são exibidas informações como o quadro do paciente assim como seu histórico.

4 O PROJETO OPEN QRBIB

4.1 Visão Geral

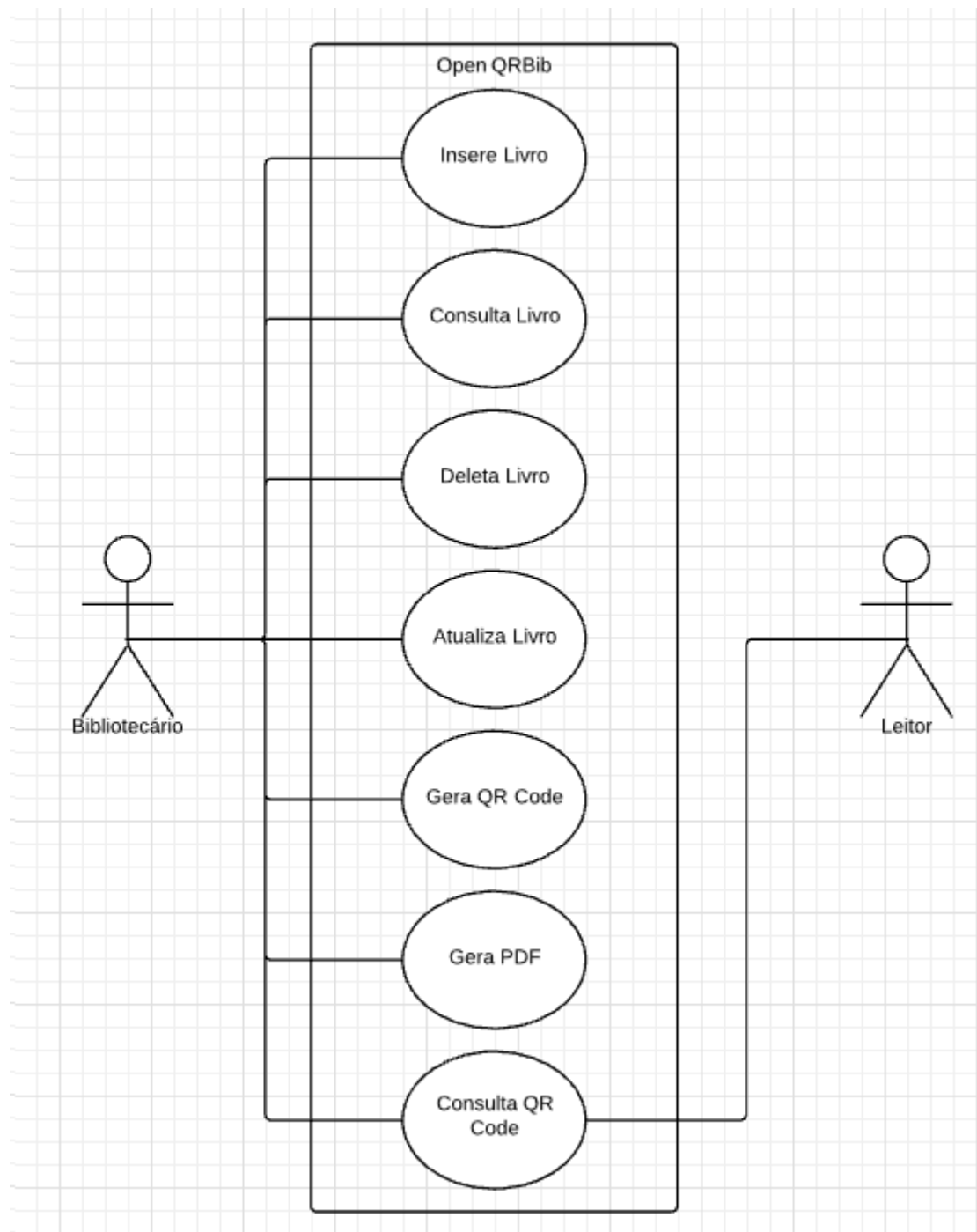
O sistema que foi desenvolvido está focado na aplicação do QR Code como fornecedor de informações de livros, possibilitando a consulta rápida de tais informações sem que seja necessária a intervenção de voluntários e ou trabalhadores das bibliotecas onde os livros estarão, ou seja, o próprio locador vai poder consultar as informações mais básicas de cada livro de uma biblioteca portando apenas um *smartphone*.

Com a Open QRBib é possível gerar múltiplos QR Codes de forma simples, onde cada QR Code gerado possuirá informações relativas a cada livro na biblioteca. Uma vez gerados cada símbolo QR será colado na contracapa de cada livro, deixando-o visível para leituras posteriores.

4.2 Descrição do software do sistema

O sistema foi planejado e projetado para que sua utilização seja feita da seguinte forma. Do lado do cliente que vai locar o livro, inicialmente, será feita a leitura do QR Code encontrado, normalmente, na contracapa dos livros, dependendo do nível de dano que o livro poderá ou não possuir, através de um dispositivo móvel (*Smartphone*), que decodificará os dados, existirão dois QR Codes em cada livro, onde o primeiro, uma vez decodificado, irá conter um *link* de acesso,

que direcionará o cliente para para uma página da internet do *TiddlyWiki*, que deve conter informações relacionadas ao livro consultado. Já o segundo QR Code foi pensado para as pessoas que por algum motivo não possuem internet móvel na hora da consulta, para essas pessoas as informações exibidas serão um pouco mais resumidas, porém ainda serão de extrema importância na hora da consulta dos livros, para facilitar o entendimento a imagem 9 a seguir mostra um diagrama de caso de uso do sistema.

Figura 9 - Diagrama de caso de uso.

Fonte: Ramos, Filho (2018)

A base de dados do Open QRBib será alimentada pelo administrador de sistema, por meio do módulo de inserção do sistema, também será função do administrador, através da Open QRBib, as seguintes tarefas relativas ao sistema, corrigir possíveis informações incorretas, atualizar as informações dos livros na

base de dados, gerar e imprimir o QR Code referente a cada livro. Fora do sistema, mas ainda dentro do escopo do projeto, também será função do administrador, administrar e alimentar as páginas da *TiddlyWiki*, para que estas se mantenham em conformidade com a base de dados e com as informações de cada livro. O escopo básico do sistema está descrito na Figura 10, a seguir:

Figura 10 - Escopo do projeto.



Fonte: Ramos, Filho (2018)

Para facilitar a impressão dos símbolos QR o sistema gerará, juntamente com os QR Codes, um arquivo no Formato Portátil de Documento, mais conhecido como PDF (do inglês *Portable Document Format*). A organização deste documento é dada em linhas e colunas, onde cada linha possuirá dois QR Codes, representando, respectivamente, o *link* para a TiddlyWiki e o QR Code que armazena as informações *offline*, também será, o título do respectivo QR Code, apresentado à sua esquerda, como mostra a Figura 11, exibida mais adiante. Como o formato PDF é um formato de arquivo que facilita a visualização do documento independentemente de qual software tenha sido utilizado para a sua criação, será mais fácil para o administrador de sistema visualizar e imprimir esse documento, sem que seja necessária posterior, edição em outros softwares, dos QR Codes para que estes sejam impressos de uma só vez.

Figura 11 - Disposição dos QR Codes no documento PDF.

	Livro_2 Link.png		Livro_2.png
	Livro_3 Link.png		Livro_3.png
	teste Link.png		teste.png

Fonte: Ramos, Filho (2018)

4.3 Principais classes na implementação do sistema

As principais classes implementadas no sistema são apresentadas abaixo, assim como uma breve explanação das suas funções:

Classe Livro - Classe responsável pela criação dos objetos livro no sistema, possui métodos *sets*, responsáveis pela alteração do valor de qualquer atributo da classe, assim como métodos *gets*, usado para acessar os valores previamente estabelecidos de determinados atributos.

Classe QRCodeConfig - Classe responsável pela criação do objeto QR Code, ela possui como atributos pontos fundamentais da configuração do QR Code final, como por exemplo o tamanho e tipo de arquivo do QR Code que deverá ser gerado. Esta classe também possui mais de um construtor, com configurações

padrões para o arquivo de saída, assim como um construtor com configurações alteráveis, para uma maior adaptação.

Classe QRMain - Esta classe tem como responsabilidade a criação do QR Code final, assim como a passagem dos parâmetros de configuração de tamanho, tipo de saída e caminho do arquivo final, através do método `configLista()`, uma lista de Livros, previamente obtida por meio de uma consulta de dados realizada pelo método `geraLivros()`, é lida e utilizando o método `geraQRCode()` os seus respectivos QR Codes são gerados.

Classe SQLiteCon - Esta classe é responsável pela configuração, criação, inserção e consulta dos dados existentes na base de dados, previamente alimentada pelo administrador do sistema, todas os acessos à base de dados são realizados por meio dessa classe.

4.4 Projeto e desenvolvimento do sistema

Para realizar o desenvolvimento do sistema se fez necessário o estudo da biblioteca ZXing, assim como uma série de estudos a respeito do próprio QR Code. O processo de aprendizado da ZXing se deu, principalmente, através do estudo de múltiplos tutoriais que explicavam o funcionamento e comportamento da biblioteca, como executar exemplos e configurar o ambiente (SINGH, [20_ _]). O processo de aprendizado sobre o QR Code se deu principalmente pelo site dos seus criadores, a Denso Wave Incorporated, onde a empresa explicava o funcionamento e a evolução da tecnologia ao longo dos anos, (DENSO, [20_ _]). Sobre a base de dados utilizada, foi escolhido o SQLite, principalmente por sua facilidade de uso e por ela ser uma boa base de dados para uso interno (APPROPRIATE, 2000). O processo de aprendizado sobre como utilizar o SQLite no escopo do projeto se deu principalmente por meio de tutoriais que ensinavam a configurar e utilizar as funções da base de dados (SQLITE, 2016).

4.4.1 Ambiente de Desenvolvimento

Os elementos mais simples para a configuração do ambiente de desenvolvimento no PC (do inglês *Personal Computer*) estão descritos abaixo:

- JVM (Java JDK SE 8);
- Eclipse IDE (do inglês *Integrated Development Environment*) Photon;
- Biblioteca ZXing 3.3.2;
- Biblioteca SQLite 3.8.11.2;
- Biblioteca Itextpdf 5.1.0.

A preparação do ambiente de desenvolvimento é feita inicialmente pela instalação da JVM (*Java Virtual Machine*) e da IDE Eclipse. Ao fim do processo de instalação da JVM e da IDE se faz necessário a importação da biblioteca utilizada para gerar e manipular os QR Codes assim como, a biblioteca responsável pela manipulação da base de dados, o SQLite, para tal importação foi utilizado o Maven, ferramenta de gerenciamento de projetos, mas, a utilização dessa ferramenta é opcional, uma vez que, é possível importar a biblioteca pelos meios tradicionais dentro da IDE, configurando o *build path*.

Os passos de importação das bibliotecas sem a utilização do Maven são os seguintes. Primeiramente, deve-se fazer o *download* dos JARs (*Java ARchive*) referentes às bibliotecas desejadas, nesse caso a ZXing, disponível em <<https://javalibs.com/artifact/com.google.zxing/core>>, a SQLite disponível em <<https://jar-download.com/artifacts/org.xerial/sqlite-jdbc/3.8.11.2/source-code>> e a Itextpdf que tem a íntegra do seu conteúdo completamente disponível em <<https://mvnrepository.com/artifact/com.itextpdf/itextpdf/5.1.0>>. Feito o *download* das bibliotecas, será necessário fazer a importação dos arquivos para o projeto, essa importação será feita acessando o menu “*Project*” e selecionando a opção “*Properties*” feito isso uma nova janela deverá aparecer na tela, na barra de pesquisa dessa janela busque “*Java Build Path*” e selecione a opção com o mesmo nome, o próximo passo é selecionar a aba “*Libraries*” e depois clicar na opção “*Add External JARs...*” feito isso uma janela para a seleção de arquivos deverá aparecer

na tela, o próximo passo é selecionar as bibliotecas que já foram baixadas previamente e clicar em “Abrir”.

4.5 O estudo das bibliotecas de geração de QR Code

O estudo dos modo de geração de QR Code teve que ser feito logo no início do projeto, pois era necessário o entendimento de como esta tecnologia funcionava, uma vez que ela é a base de todo o sistema. Após os estudos foram realizadas buscas em cima de bibliotecas já prontas, que oferecessem meios de para a geração desses códigos dinamicamente, através do uso da linguagem de programação Java.

Dentre as bibliotecas encontradas duas se destacaram, e foram selecionadas para um estudo mais aprofundado, foram elas a ZXing e a QRGen.

Após uma análise mais aprofundada foi verificado que a QRGen poderia ser uma ótima candidata para o projeto, pois ela possuía uma linguagem relativamente simples e de fácil entendimento, porém após mais algum tempo de estudos e análises, foi verificado que o suporte da mesma não era tão bom, mas o que fez com que ela fosse definitivamente descartada definitivamente foram os fatos dela, ser uma Interface de programação de aplicações ou API (do inglês *Application Programming Interface*) montada em cima da biblioteca ZXing e da complicação encontrada para criar aplicações que não fossem nativas do sistema Android, uma vez que a aplicação do projeto se destina à dispositivos desktops tal biblioteca não se enquadrou de forma eficaz ao escopo.

A segunda biblioteca de criação de códigos QR que foi estudada de forma mais aprofundada foi a ZXing. Esta biblioteca, apesar de ser um pouco mais complexa se comparada com a QRGen, se mostrou a melhor escolha para a aplicação no projeto, pois a mesma apresentava um ótimo suporte para eventuais dúvidas no decorrer do projeto, ela também se mostrou uma ótima escolha para aplicações desktop, além de apresentar uma gama maior de opções para a criação e configuração dos símbolos.

4.6 Prova de conceito do sistema

Nesta seção serão apresentadas algumas partes do código fonte do projeto, em sua maioria métodos importantes, codificados para auxiliar a geração assim como gerar os QR Codes pertinentes à aplicação do projeto como um todo, além disso, também são mostradas algumas telas do próprio sistema.

A Figura 12 mostra o método *configLista*, responsável por configurar a lista de livros recebida da base de dados para a posterior geração do símbolo QR Code, também é nele que os nomes dos símbolos serão configurados e há a divisão de criação dos QR Codes que contém o *link* para a TiddlyWiki, e dos QR Codes que possuem as informações *offline* dos livros existentes tanto no sistema quanto na biblioteca, por último, mas não menos importante, este método também define o diretório onde os QR Codes deverão ser criados, criando um, caso o diretório definido não exista previamente.

Figura 12 - Método de configuração de lista de livros.

```
public void configLista() {
    String homeDoUsuario = System.getProperty("user.home");
    String filePath = homeDoUsuario + "/Documents/QRCodes/";
    new File (filePath).mkdirs();

    int geraLink = 0;

    for (Livro livro : lLivros) {
        QRconfig = new QRCodeConfig(filePath + livro.getNome());
        String myCodeText = CodeText(livro, geraLink);
        GeraQRCode(myCodeText);
        geraLink++;

        QRconfig = new QRCodeConfig(filePath+ livro.getNome() + " Link");
        myCodeText = CodeText(livro, geraLink);
        GeraQRCode(myCodeText);
        geraLink = 0;
    }
}
```

A Figura 13, mostrada mais abaixo, exibe o método geraPdf responsável pela geração do arquivo PDF, utilizado para impressão dos QR Codes que posteriormente serão fixados nos livros.

Figura 13 - Método de geração de PDF

```
private void geraPdf(String filePath) {
    new File (filePath + "PDF/").mkdirs();

    File folder = new File(filePath);
    File[] listOfFiles = folder.listFiles();

    String FILE_NAME = filePath + "PDF/QRCodes.pdf";
    Document document = new Document();
    try {
        PdfWriter.getInstance(document, new FileOutputStream(new File(FILE_NAME)));
        document.open();

        PdfPTable table = new PdfPTable(4);
        table.setWidthPercentage(100);
        table.setWidths(new int[]{2, 1, 2, 1});

        for (File file : listOfFiles) {
            if(file.isFile() && file.getName().toLowerCase().endsWith(".png")) {
                Image qr = Image.getInstance(filePath+"/"+file.getName());
                table.addCell(qr);
                table.addCell(file.getName());
            }
        }
        document.add(table);
        document.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Fonte: Ramos, Filho (2018)

A figura 14 mostra, o que é possivelmente, o núcleo do sistema, pois ela mostra o método geraQRCode, como o próprio nome do método sugere, ele é responsável pela geração de cada QR Code que o sistema produzirá, ele utiliza as configurações previamente estabelecidas pela classe QRConfig, para criar os QR Codes padronizados em tamanho e formato.

Figura 14 - Método de geração dos QR Codes.

```

public void GeraQRCode(String myCodeText) {

    int size = QRconfig.getSize();
    String fileType = QRconfig.getFileType();
    File myFile = QRconfig.getMyFile();
    try {
        Map<EncodeHintType, Object> hintMap = new EnumMap<EncodeHintType, Object>(EncodeHintType.class);
        hintMap.put(EncodeHintType.CHARACTER_SET, "UTF-8");

        hintMap.put(EncodeHintType.MARGIN, 1); /* default = 4 */
        hintMap.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.L);

        QRCodeWriter qrCodeWriter = new QRCodeWriter();
        BitMatrix byteMatrix = qrCodeWriter.encode(myCodeText, BarcodeFormat.QR_CODE, size,
            size, hintMap);
        int largura = byteMatrix.getWidth();
        BufferedImage image = new BufferedImage(largura, largura,
            BufferedImage.TYPE_INT_RGB);
        image.createGraphics();

        Graphics2D graphics = (Graphics2D) image.getGraphics();
        graphics.setColor(Color.WHITE);
        graphics.fillRect(0, 0, largura, largura);
        graphics.setColor(Color.BLACK);

        for (int i = 0; i < largura; i++) {
            for (int j = 0; j < largura; j++) {
                if (byteMatrix.get(i, j)) {
                    graphics.fillRect(i, j, 1, 1);
                }
            }
        }
        ImageIO.write(image, fileType, myFile);
    } catch (WriterException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Fonte: Ramos, Filho (2018)

A figura 15 mostra a tela inicial do sistema, a partir dela o usuário será capaz de acessar as subopções e navegar pelo sistema como um todo, além disso a partir dessa tela é possível gerar os QR Codes e o PDF de todos os livros do sistema.

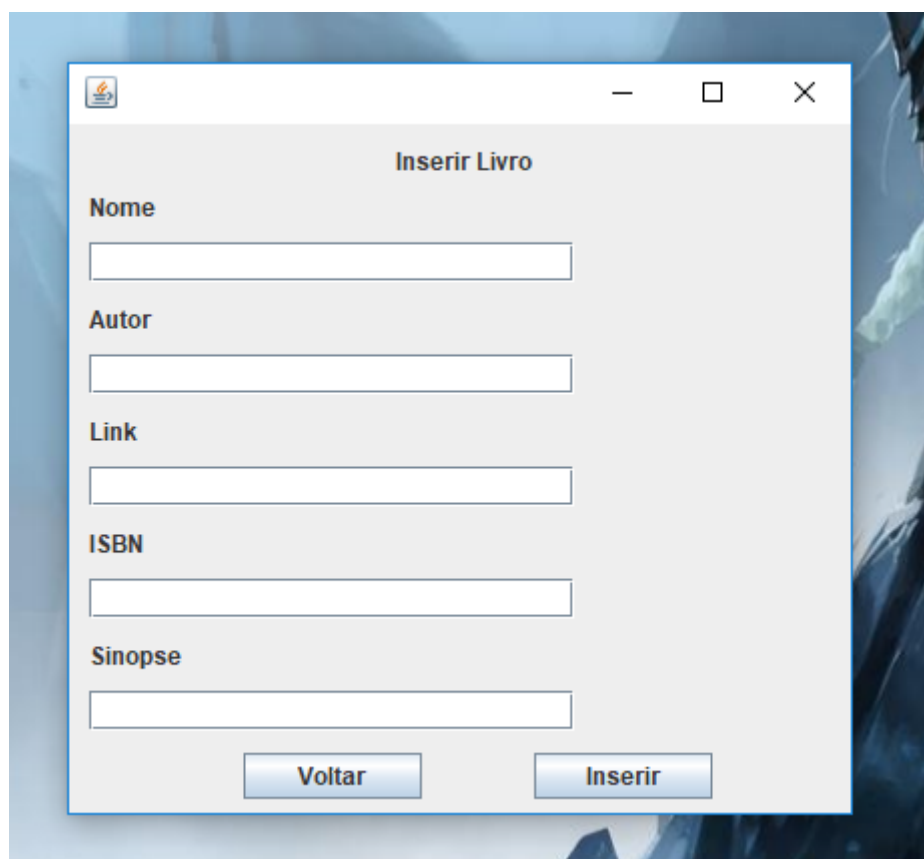
Figura 15 - Tela inicial do sistema.



Fonte: Ramos, Filho (2018)

Na figura 16 é exibida a tela de inserção de novos livros no sistema, ela deverá receber cinco parâmetros do usuário, como é mostrado mais abaixo.

Figura 16 - Tela de inserção de livros.



A imagem mostra uma janela de aplicativo com o título "Inserir Livro". O formulário contém os seguintes campos:

- Nome
- Autor
- Link
- ISBN
- Sinopse

Na base da janela, há dois botões: "Voltar" e "Inserir".

Fonte: Ramos, Filho (2018)

A figura 17 é relativa a consulta dos livros já inseridos, nela será possível pesquisar os livros a partir de três parâmetros que deverão ser fornecidos individualmente, além disso também é possível fazer uma consulta de todos os livros já existentes no acervo.

Figura 17 - Tela de consulta de livros.

The screenshot shows a window titled 'Consulta de livros' with three search options: 'Consultar por Nome:', 'Consultar por ISBN:', and 'Consultar por Autor:'. Each option has a corresponding text input field. At the bottom, there are three buttons: 'Voltar', 'Consultar', and 'Consulta Completa'.

Fonte: Ramos, Filho (2018)

A figura 18 é relativa ao resultado retornado da tela de consulta, nela será possível visualizar os resultados da consulta, gerar os QR Codes referentes à cada livro retornado pela consulta e alterar os registros já existentes no sistema.

Figura 18 - Resultado da consulta.

ID	Nome	Autor	Link	ISBN	Sinopse	Gerar QRCode	Alterar Registro
1	Livro_1	Efraim	https://www.go...	65687	blabla	Gerar	Alterar
2	Livro_2	Jacy	https://www.bl...	65688	blablabla	Gerar	Alterar
3	Livro_3	Silvia	https://www.fa...	65689	blablablabla	Gerar	Alterar
4	Contos	Silvia	https://www.fa...	65689	blablablabla	Gerar	Alterar
5	A louca	efraim	alouca.com	fala de uma d...	Gerar	Alterar
6	teste	teste	www.teste.co...	teste	teste	Gerar	Alterar
7	Funcionou	Funcionou	www.funciono...	Funcionou	Funcionou	Gerar	Alterar
8	bla bla	Marcos	www.blabla.co...	68531	Um Livro de te...	Gerar	Alterar
9	Efraim	Efraim	www.efraim.co...	26358	Um livro de te...	Gerar	Alterar
10	video	video	www.video.co...	52698	um video de te...	Gerar	Alterar

Fonte: Ramos, Filho (2018)

5 CONSIDERAÇÕES FINAIS

Com esse trabalho foi possível aprender novas tecnologias, e visitar outras, de modo que o aprendizado obtido foi muito válido.

Com relação ao QR Code foi possível entender melhor a sua complexidade, em particular, suas múltiplas aplicações nos mais diferentes e variados campos da ciência.

Habilidades de programação puderam ser melhoradas. Principalmente após a manipulação de *hash maps*, de interfaces gráficas e arquivos, a manipulação desses exigiu uma melhora significativa das habilidades.

Também foi possível verificar que, apesar do sistema ter uma carga de trabalho inicial relativamente alta, dependendo do acervo da biblioteca, ele ainda assim, é viável para a aplicação, por ser capaz de recuperar informações perdidas, de forma rápida e simples. Além disso uma maior autonomia será dada aos leitores, uma vez que a consulta poderá ser realizada por eles mesmos, sem que exista a interferência de terceiros, de modo rápido e simples.

Durante o desenvolvimento do sistema, alguns problemas foram identificados. A ideia inicial do projeto era a sua aplicação em grandes livrarias e bibliotecas, no entanto, após observações em tais locais, verificou-se certa resistência da aplicação do projeto em larga escala, principalmente por conta do enorme trabalho manual necessário para a fixação dos símbolos QR nas vastas prateleiras desses locais, assim como a falta de danos nos livros, tornando a obtenção das informações simples, sem contar que, normalmente, tais estabelecimentos, possuem um sistema de terminais onde é possível a leitura do código de barras que já vem impresso de fábrica no verso dos livros. Diante desta barreira da aplicação nesses locais, foi verificado que as bibliotecas comunitárias não possuíam tal sistema de terminais, nem uma quantidade de livros que impossibilitasse a aplicação manual dos códigos, além disso foi verificado que, devido a obtenção de livros desse tipo de biblioteca ser praticamente todo por meio de doações, muitos dos livros encontrados estão desgastados e/ou danificados de

alguma forma, esse fato, abriu uma nova oportunidade para a aplicação da Open QR Bib em um novo ambiente.

Um outro problema encontrado também, foi com relação a leitura dos códigos que armazenavam URLs (do inglês *Uniform Resource Locator*), traduzida como Localizador Uniforme de Recursos, e texto simultaneamente, pois a leitura do QR code apenas identificava um único tipo de informação, em outras palavras, caso houvesse apenas uma URL o código era lido como um *link*, porém caso a URL estivesse acompanhada por textos o *link* era ignorado e todas as informações decodificadas eram tratadas exclusivamente como texto. Este problema foi resolvido com a adição de um novo símbolo QR que armazenava apenas o *link* da TiddlyWiki, um ponto bom que este problema nos trouxe foi a inclusão de pessoas que não possuem internet na hora da consulta, pois com a criação de múltiplos códigos tanto pessoas que possuem internet móvel, como pessoas que não possuem, podem ser beneficiadas com o uso dos QR Codes.

Também foram encontrados problemas com a aplicação do sistema na biblioteca comunitária de teste, pois a mesma não possuía um computador, em conversa com o administrador da mesma, ele disse que um usuário havia prometido fazer a doação de um computador para a biblioteca no entanto, ele ainda não havia chegado até o momento da finalização deste trabalho, por conta disso, a aplicação do sistema na sua efetividade total não pôde ser realizada.

Muito foi aprendido e lembrado no desenvolvimento deste trabalho. Os próximos passos são a aplicação efetiva do projeto em uma biblioteca comunitária a fim de observar as vantagens proporcionadas pelo mesmo de forma efetiva, dando continuidade e melhorando o desenvolvimento do sistema por meio do *feedback* fornecido pelos leitores e principalmente pelos próprios bibliotecários. Com esse *feedback* será possível a elaboração de testes mais extensos e específicos para avaliar o funcionamento e a aplicação do sistema, assim como a aceitação do mesmo por parte dos usuários.

REFERÊNCIAS

BÔAS, Rodrigo Vilas. QR Code possui diversas aplicações principalmente na área da publicidade. **A Tarde**, 2008. Disponível em: <<http://atarde.uol.com.br/bahia/salvador/noticias/1279001-qr-code-possui-diversas-aplicacoes-principalmente-na-area-da-publicidade>>. Acesso em: 24 out. 2018.

DENSO Wave Incorporated. **Information capacity and versions of the QRCode**, [20_ _]. Disponível em: <<http://www.qrcode.com/en/about/version.html>>. Acesso em: 31 out. 2018.

DENSO Wave Incorporated. **Denso Wave, the inventor of QR Code**, [20_ _] Disponível em <<http://www.qrcode.com/en/index.html>>. Acesso em: 28 set. 2018.

DENSO Wave Incorporated. **What is a QR Code**, 2018, [20_ _] Acesso em 08 nov. 2018. Disponível em: <<http://www.qrcode.com/en/about/>>

RAPOSO, Alberto. Conheça o QRCode, o código que tomou conta do mundo mobile. **Globo Ciência**, 2013. Disponível em: <<http://redeglobo.globo.com/globociencia/noticia/2013/05/conheca-o-qr-code-o-codigo-que-tomou-conta-do-mundo-mobile.html>>. Acesso em: 31 out. 2018.

INFORMATION Technology Standards Committee. QR Code. **Synthesis Journal**, Singapura, seção 3, p. 59 - 78. Disponível em: <https://foxdesignsstudio.com/uploads/pdf/Three_QR_Code.pdf>. Acesso em: 28 out. 2018.

INTERNATIONAL Organization for Standardization (ISO). **ISO/IEC 18004:2015. Information technology -- Automatic identification and data capture techniques -- QR Code bar code symbology specification**. 3 ed: [S./], IEC, 2015.

RAMOS, Efraim de Souza; CHAGAS, Jorge José Espírito Santo; SANTOS, Taysa Ferreira dos. **Implantação de QR-Code para Turismo no Cemitério de Santo Amaro**. Recife: [s.n.], 2018. p. 19-27.

SINGH, Rajeev Kumar. **Generate QR Code in java using zxing**, [20_ _]. Disponível em: <<https://www.callicoder.com/generate-qr-code-in-java-using-zxing/>>. Acesso em: 12 set. 2018.

SQLITE Java. **SQLite Tutorial**, 2016. Disponível em: <<http://www.sqlitetutorial.net/sqlite-java/>>. Acesso em: 21 set. 2018.

APPROPRIATE Uses For SQLite. **SQLite**, 2000. Disponível em: <<https://www.sqlite.org/whentouse.html>>. Acesso em: 20 set. 2018.