INTERFACE TECNOLÓGICA

INTEGRAÇÃO ANDROID COM SAP UTILIZANDO WEBSERVICES

ANDROID INTEGRATION WITH SAP USING WEBSERVICES

Gabriel Vendramini Bueno – gabrielvbueno@uniara.edu.br

Rodrigo Daniel Malara – rmalara.uniara@gmail.com

Fabiana Florian – eco_fab@hotmail.com

Universidade de Araraquara (UNIARA) – SP – Brasil

DOI: 10.31510/infa.v16i2.671

RESUMO

Enterprise Resource Planning (ERP) são *softwares* que integram informações de todos os setores da empresa em um único sistema. O SAP é um sistema ERP de alta complexidade que permite manter um ótimo planejamento e controle do negócio. O objetivo deste artigo é desenvolver um aplicativo cujo propósito é executar a integração entre os sistemas SAP e dispositivos Android. O crescimento do número de usuários de dispositivos Android permitiu que este sistema operacional fosse escolhido para demonstração desta integração. O aplicativo desenvolvido permite visualizar: dados de notas fiscais vindas do monitor de notas do SAP; dados cadastrais de clientes e fornecedores tornando possível o acesso à telefones, endereços e dados bancários; e indicadores para auxílio na tomada de decisões em processos importantes como compras e vendas de produtos, controle de produção, entre outras atividades. Conclui-se que foi possível demonstrar que os dados do SAP podem ser exibidos em dispositivos móveis.

Palavras-chave: Android. ERP. Kotlin. SAP. WebServices.

ABSTRACT

Enterprise Resource Planning (ERP) is software that integrates information from all sectors of the company into a single system. SAP is a highly complex ERP system that allows you to maintain optimal business planning and control. The purpose of this article is to develop an application whose purpose is to perform integration between SAP systems and Android devices. The growing number of Android device users has allowed this operational system to be chosen to demonstrate this integration. The application developed allows you to view: invoice data coming from the SAP invoice monitor; customer and supplier registration data making it possible to access telephones, addresses and bank details; and indicators to assist in decision making in important processes such as product purchasing and sales, production control, among other activities. It follows that it was possible to demonstrate that SAP data can be displayed on mobile devices.

Keywords: Android. ERP. Kotlin. SAP. WebServices.

1 INTRODUÇÃO

Enterprise Resource Planning (ERPs) que em português significa Planejamento de Recursos Corporativos são sistemas corporativos que unem vários microssistemas, sendo um produto completo para gerenciamento de uma empresa, disponibilizando várias funções, e por possuir muito conteúdo, algumas destas funções podem nunca ser utilizadas. Este tipo de sistema permite a integração de dados e informações de diversos setores da empresa.

O SAP é um ERP que possui diversos módulos de atuação que disponibilizam diversas soluções para todos os setores de uma empresa. Existe um produto da SAP chamado Fiori, que permite utilização de soluções do cliente SAP em telas do navegador *Web* e em um aplicativo *mobile*, tornando algumas funcionalidades mais dinâmicas.

O objetivo deste artigo é desenvolver um aplicativo cujo propósito é executar uma alternativa de integração das soluções SAP com dispositivos Android, utilizando a linguagem de programação Kotlin. A hipótese desta pesquisa é que a utilização do consumo de dados via *WebService* permitirá exibir relatórios e visualizar dados importantes. Para demonstrar esta integração, é desenvolvido um aplicativo denominado "*SAP Service Portable"* (SSP) que consiste em uma ferramenta para visualização de dados do SAP no Android. Também são apresentadas informações sobre as tecnologias envolvidas no processo de integração entre o SAP e o Android.

O Android já é o sistema operacional mais usado do mundo, conta com 37,93% de usuários enquanto o Windows, plataforma da Microsoft alcança 37,91% (ELPAIS, 2017), e a quantidade de usuários permitiu que este fosse o sistema operacional escolhido para demonstração da integração de dados do SAP com dispositivos móveis.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta sessão são apresentadas informações relacionadas a tecnologias utilizadas para demonstração do funcionamento da integração do SAP com aparelhos Android.

2.1 Arquitetura Orientada à Serviços (SOA)

Para otimização e maior controle de processos, empresas utilizam sistemas diferentes para diferentes áreas de negócios. O ERP (tipo de sistema utilizado neste artigo para integração com Android) é utilizado para controlar sua operação; CRM (*Customer Relationship Management*) para gerenciar interações com clientes; e o SCM (*Supply Chain Management*) para gerenciar interações com parceiros. Com este tipo de estrutura, onde cada sistema possui seu próprio conjunto de informações, é possível que exista a necessidade de troca de informação entre estes sistemas, cenário no qual a Arquitetura Orientada a Serviços (SOA) se encaixa (HIRAMA e FUGITA, 2012).

Service Oriented Architecture (SOA) é um paradigma de desenvolvimento de aplicações cujo objetivo é criar módulos funcionais chamados de serviços, permitindo a reutilização de código (SAMPAIO, 2006). No contexto de arquitetura de *software*, o conceito de serviço está ligado à capacidade de um sistema executar uma função para outro sistema (HIRAMA e FUGITA, 2012) atendendo uma função de negócio específica, recebendo requisições e respondendo-as ocultando todo o detalhamento do seu processamento (SAMPAIO, 2006).

Serviços são compostos por várias operações, de forma semelhante à metodos de classes em orientação a objetos, também possuindo parâmetros de entrada e de saida. Ao consumir um serviço, o cliente envia uma mensagem com parâmetros de entrada, então, o provedor do serviço executa a lógica da operação, e retorna com uma mensagem com parâmetros de saída. Serviços possuem uma característica chamada granularidade, que pode ser baixa (técnico ou de baixo nível) ou alta (serviço de negócio ou alto nível) variando de acordo com a quantidade de funcionalidades encapsuladas, ou seja, quanto mais funcionalidades um serviço possuir, mais alta é sua granularidade. Serviços que apoiarão processos de negócio devem ser definidos de forma adequada e devem ser especificados para que possam fazer parte de um repositório de serviços reusáveis (HIRAMA e FUGITA, 2012).

2.2 WebService

Um *WebService* Pode ser definido como um componente de *software* que possui uma interface separada da sua implementação, expondo para seus consumidores apenas a interface ocultando o detalhamento do processamento (HIRAMA e FUGITA, 2012). Segundo a SAP, *WebServices* podem ser definidos da seguinte forma:

> Um Web Service é uma função ou serviço de aplicativo independente, modular e autodescritivo. Com base em XML e outros padrões, essa função de aplicativo pode ser descrita, disponibilizada, localizada ou chamada usando protocolos da Internet. Cada serviço da Web, portanto, encapsula uma parte da funcionalidade que pode ser usada, por exemplo, para encaminhar uma consulta de preço a um provedor, verificar a disponibilidade de um item em um sistema de planejamento de recursos corporativos ou localizar um número de telefone (SAP, 2019, p. 1).

O protocolo utilizado para troca de informação entre o serviço e o cliente é o *Simple Object Access Protocol* (SOAP) ou Protocolo Simples de Acesso a Objetos. SOAP utiliza as tecnologias *Extensible Markup Language* (XML) ou linguagem de marcação extensível e *HyperText Transfer Protocol* (HTTP) ou Protocolo de Transferência de Hipertexto para consumir métodos de componentes remotos. O protocolo SOAP utiliza dois métodos para executar requisições em serviços: GET e POST, e cada um pode ser utilizado em um tipo de situação. O método GET é utilizado para requisições idempotentes, ou seja, podem ser repetidos com os mesmos resultados diversas vezes, por exemplo, buscando dados para exibição. O método POST é utilizado quando as requisições alteram o estado do servidor, e não podem ser repetidas sem a ocorrencia de problemas, como por exemplo, um formulário de compras *on-line* (SAMPAIO, 2006).

2.3 XML

Ao executar a busca de dados via *WebService*, os dados são retornados no formato XML e tratados em código utilizando a linguagem Kotlin. O XML é uma representação de dados em forma de texto que possui como componente básico os “elementos” (ou tags) que são textos intercalados pelos caracteres “<” e “>”, como, por exemplo as tags, “<item>” e “</item>”. Dentro do elemento tem-se texto bruto, outro elemento ou uma mistura dos dois.

Uma expressão <item> é chamada marcação inicial, e </item> é chamada marcação final. Estas marcações são definidas pelo usuário. O texto entre a marcação inicial e a final, inclusive as marcações, é chamado “elemento”; os dados localizados entre as marcações são descritos como o “conteúdo”, como no exemplo “<item>xml_item</item>”, os caracteres entre as marcações (“xml_item”) são o conteúdo do elemento. O termo “subelemento” é também usado para descrever a relação entre um elemento e os outros elementos que o compõem (ALMEIDA, 2002).

2.4 Android

Android é uma plataforma para tecnologia móvel completa, já com sistema operacional, *middleware*, aplicativos e interface do usuário, e, por ser *open source*, ou seja, de código aberto, pode ser sempre adaptado a fim de incorporar novas tecnologias conforme forem surgindo. Esse sistema operacional foi construido com base no Linux, porém, não possui algumas características do sistema no qual foi baseado (PEREIRA e SILVA, 2009). Além de *smartphones*, o sistema operacional Android pode ser utilizado em outros dispositivos, como televisões. Uma das maiores vantagens do Android é sua biblioteca de aplicativos, que em seu catálogo, possuem diversos tipos, como aplicativos para entretenimento, trabalho, aumento de produtividade e etc.

O Sistema Android possui uma arquitetura dividida em camadas (Figura 1).

Figura 1 - Camadas da arquitetura do sistema operacional Android

Fonte: PEREIRA e SILVA (2009)

A camada de Aplicação (*Applications*) é onde são executados os aplicativos fundamentais como um cliente de e-mail, mapas, navegadores, calendários, agendas, entre outros. O Quadro de Aplicações (A*pplication Frameworks*), serve de suporte para o funcionamento da camada de Aplicações. Nele estão dispostas todas as APIs e os recursos utilizados pelos aplicativos incluindo componentes visuais como botões, caixas de texto, e etc. Alguns dos elementos desta camada são: *Activity Manager*, *Window Manager*, *Content Provider, View System* (PEREIRA e SILVA, 2009).

O *Activity Manager* gerencia o ciclo de vida das activities; o *Window Manager* executa o gerenciamento das janelas; o *Content Provider*, possibilita o compartilhamento de dados entre aparelhos; e o *View System* é responsável por disponibilizar os elementos gráficos como botões e caixas de texto (PEREIRA e SILVA, 2009).

Após a camada do quadro de aplicações (A*pplication Frameworks* – Figura 1*)*, há a camada de Bibliotecas (*Libraries*), que possui várias bibliotecas das linguagens de programação C/C++ utilizadas por vários componentes do sistema operacional. Além das bibliotecas C/C++, esta camada também possui bibliotecas da área multimídia, visualização 2D e 3D, funções para navegadores *Web*, funções para gráficos, funções de aceleração de *hardware*, renderização 3D, fontes bitmap e vetorizadas e funções de acesso ao banco de dados *SQLite*. Dentre as principais bibliotecas, destacam-se: o *Free Type*, responsável por renderização de fontes e bitmaps; *o WebKit*, responsável por renderizar páginas *Web* com suporte à CSS; e o *SQLite*, uma engine de banco de dados relacional implementada em C, leve e embutida com suporte a uma base de dados acima de 2 terabytes (PEREIRA e SILVA, 2009).

Outra camada do sistema operacional Android é a camada do ambiente de execução (*Android Runtime*) que possui pacotes básicos Java para um ambiente quase completo de programação e funciona como uma instância da máquina virtual utilizada pelo Android chamada de Dalvik. Essa máquina virtual permite maior integração com novas gerações de *hardware* e foi projetada para executar várias máquinas virtuais paralelamente. A Dalvik executa arquivos com a extensão DEX (*Dalvik Executable*) que são classes de Java convertidas para a máquina virtual através da ferramenta DX, distribuída com o SDK (*Software Development Kit* - pacote de desenvolvimento de *software*) do Android (PEREIRA e SILVA, 2009).

A última camada é a *Linux Kernel*, que atua como uma camada de abstração entre o *hardware* e o resto da pilha de *software*. Esta camada é utilizada para executar o gerenciamento de serviços centrais do sistema, como gestão de memória (PEREIRA e SILVA, 2009) e promete agilidade e portabilidade para tirar vantagem das diversas opções de *hardware* de futuros telefones Android.

Android possui seu código aberto, e utilizar uma fundação de código-fonte aberto libera as capacidades de companhias e indivíduos talentosos para que a plataforma esteja sempre em aprimoramento (ABLESON, SEN, KING e ORTIZ, 2012).

2.5 Kotlin

Aplicativos Android podem ser criados com diversas linguagens de programação, como Java, Kotlin e .NET.

> [...]Kotlin é uma linguagem concisa, segura, pragmática, com enfoque na interoperabilidade com código Java. Ela pode ser usada em quase todos os lugares em que Java é utilizada atualmente: em desenvolvimentos do lado do servidor, aplicações para Android e muito mais. Kotlin funciona muito bem com todas as bibliotecas e os frameworks Java desenvolvidos e executa apresentando o mesmo nível de desempenho que Java (JAMEROV; ISAKOVA, 2017, p. 3).

As aplicações mais comuns para o Kotlin são: construir código do lado do servidor incluindo aplicações *Web*; e criar aplicativos móveis que são executados em dispositivos Android. Também pode ser utilizado em outros contextos, por exemplo, com o *Intel Multi-OS* é possível executar códigos em Kotlin em dispositivos com o sistema operacional IOS.

Kotlin é uma linguagem pragmática, concisa, segura e com foco em interoperabilidade. A linguagem pragmática é projetada para resolver problemas do mundo real; a consisa possui sintaxe que expressa claramente a intenção do código; a segura indica que seu design previne certos tipos de erros em um programa, porém, evitar erros tem um custo, pois é necessário fornecer ao compilador mais informações sobre a operação pretendida, por conta disso, pode haver um conflito na escolha do nível de segurança. E por fim, Kotlin é uma linguagem com foco em interoperabilidade, o que indica que classes e métodos Kotlin podem ser chamados como classes e métodos Java, provendo uma ótima flexibilidade na mistura de código Java e Kotlin em qualquer parte do projeto (JAMEROV e ISAKOVA, 2017).

2.6 SAP

O SAP (Sistemas, Aplicativos e Produtos para Processamento de Dados) é um sistema ERP de alta complexidade que permite manter um ótimo planejamento e controle do negócio. O R/3 (Produto da SAP) mantém todos os sistemas unidos e é a espinha dorsal do sistema geral (DAVENPORT, 2002).

> O sistema SAP R/3 é composto por um conjunto de módulos de software integrados iterativamente. É um sistema abrangente e complexo. Pode tratar atividades desde a cadeia produtiva até relacionamentos com clientes da organização. As aplicações partilham dados de bases comuns aos módulos. As alterações feitas em bases de dados por determinado programa aplicativo não compromete a funcionalidade de outros módulos do sistema. A implementação do SAP R/3 em qualquer empresa

> requer a análise e a modificação de formas de realização de processos do negócio (SANTOS; KALDEICH; SILVA, 2003, p. 3).

O SAP R/3, assim como o sistema operacional Android, possui uma arquitetura dividida em camadas. No SAP, a divisão é feita em três camadas: camada de dados; camada de aplicação; e camada de apresentação. A de dados é responsável por prover dados ao sistema, incluindo tabelas, dados de transações e dados da empresa incluídos no sistema; a de aplicação é responsável pelo processamento e intercâmbio de dados entre o servidor de aplicação e o servidor de base de dados para a execução de ferramentas do *software*; e a de apresentação atua como interfaces (*front-end*) de terminais e recursos de usuários (SANTOS, KALDEICH e SILVA, 2003).

3 PROCEDIMENTOS METODOLÓGICOS

Além da realização da pesquisa bibliográfica para demonstrar as tecnologias envolvidas na integração entre o SAP e o Android, foi executado o desenvolvimento de um aplicativo para exibição de dados do SAP com o intuito de validar informações e facilitar a tomada de decisões. Este aplicativo foi desenvolvido utilizando a ferramenta *Android Studio*, utilizando a linguagem Kotlin.

O primeiro passo para execução do desenvolvimento da integração das soluções SAP com dispositivos Android foi a criação de *WebSevices* no ambiente SAP. Estes *WebServices* foram desenvolvidos utilizando a ferramenta de criação disponibilizada pela SAP. Para utilização desta ferramenta foi necessário desenvolver um módulo de função (trechos de código reutilizáveis globalmente no ambiente SAP) de acesso remoto para cada uma das funcionalidades criadas e então, a ferramenta disponibiliza um WSDL com as informações para que seja feito o consumo do serviço. Os códigos desenvolvidos nestes *WebServices* executam seleções e tratamento de dados para que seja feita a exibição no aplicativo.

Após o desenvolvimento dos *WebServices*, o próximo passo foi o desenvolvimento das telas do aplicativo (ou *activities*). As principais telas desenvolvidas foram: tela de *login*; indicadores; notas fiscais; e parceiros. Cada tela possui uma função específica, e todas buscam apresentar informações pertinentes utilizando os serviços gerados anteriormente no SAP. Os serviços são consumidos utilizando o *ksoap2*, um *framework* que fornece uma biblioteca SOAP leve e eficiente para a plataforma Android. Ao consumir os serviços, os parâmetros de

entrada do mesmo são enviados via XML, e então, quando encontrados, os dados são retornados também no formato XML, então, são tratados utilizando DOM Parser.

Para executar a validação da veracidade dos dados exibidos nas funcionalidades do aplicativo foram feitas verificações nas tabelas do banco de dados do SAP utilizando a ferramenta para visualização de dados do banco do próprio SAP.

4 RESULTADOS E DISCUSSÃO

Além da tela de *login*, as três principais telas criadas para consumo de dados no aplicativo desenvolvido são as telas de indicadores, notas fiscais e parceiros. Estas telas são utilizadas para definição dos parâmetros para a seleção de dados, e após a execução, as telas com os dados solicitados são exibidas.

4.1 Tela de login

Na tela de *login* (Figura 2) ocorre a validação de usuário e senha. O usuário deve ser o mesmo utilizado para executar o *login* no ambiente do SAP, ou seja, não exige um cadastro prévio de usuários por parte da equipe responsável.

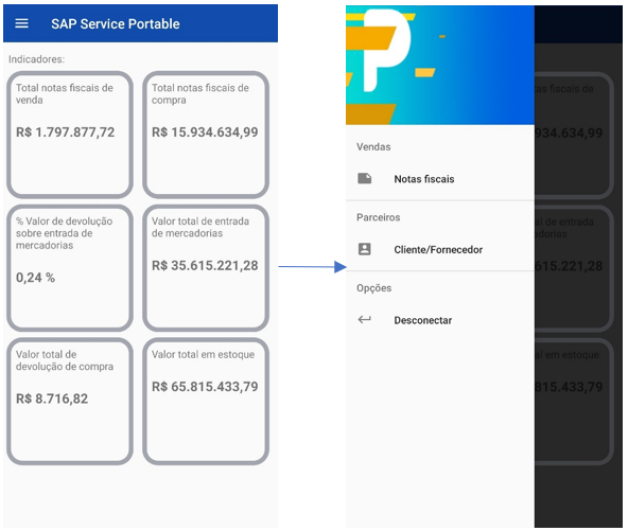
Figura 2 – Tela de Login

Fonte: Os autores (2019)

4.2 Tela de indicadores

A tela de indicadores (Figura 3) é a tela inicial do aplicativo após a tela de *login*, e possui informações para auxílio na tomada de decisões e um menu lateral que direciona para outras telas ou para a opção de desconectar.

Figura 3 – Tela de indicadores e menu lateral

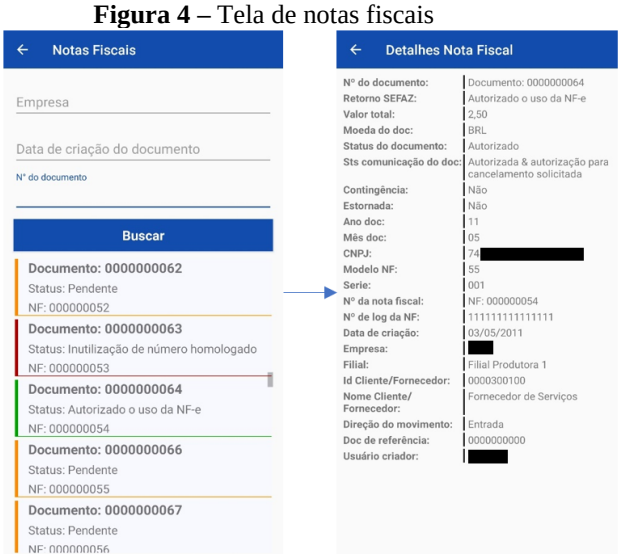


Fonte: Os autores (2019)

4.3 Tela de notas fiscais

A tela de notas fiscais (Figura 4) possui três parâmetros para seleção (empresa, data de criação da nota fiscal e número da nota fiscal) de notas ficais que são enviados para o *WebService* e uma lista exibida abaixo destes parâmetros quando os dados são obtidos. Notas fiscais com *status* pendente estão contornadas com a cor laranja, enquanto as aprovadas ficam destacadas em verde e as reprovadas estão destacadas com vermelho. Ao tocar em qualquer nota na lista, uma tela de detalhes da nota é exibida, com informações como número da nota fiscal, usuário criador, data de criação, e outros dados relacionados à nota fiscal.

Figura 4 – Tela de notas fiscais

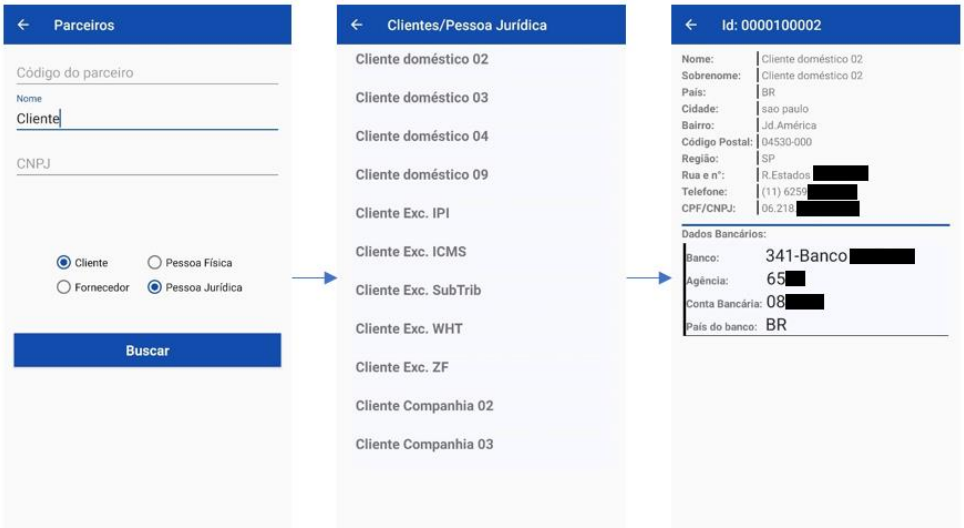


Fonte: Os autores (2019)

4.4 Tela de parceiros

A tela de parceiros (Figura 5) possui características semelhantes à tela de seleção de notas fiscais, com as opções para buscar clientes ou fornecedores que são pessoas físicas ou jurídicas. Ao encontrar dados, outra tela é exibida com nomes dos parceiros, também com a possibilidade de tocar em qualquer item da lista exibindo outra tela com características como endereço, telefone e dados bancários.

Figura 5 – Tela de busca de parceiros, tela de exibição de parceiros e tela de detalhes do parceiro



Fonte: Os autores (2019)

5 CONSIDERAÇÕES FINAIS

Com o desenvolvimento do aplicativo, foi possível demonstrar que os dados do SAP podem ser exibidos em dispositivos móveis, portanto, o objetivo proposto neste artigo foi alcançado. *WebServices* podem ser utilizados em combinação com diversas linguagens de programação, o que torna possível que a integração criada neste artigo possa ser feita com outros tipos de aparelhos, executando um desenvolvimento *Web* para que a aplicação desenvolvida possa ser executada no navegador de qualquer aparelho ou utilizando tecnologias como Xamarin (Plataforma mantida pela Microsoft que permite a criação de aplicativos para IOS, Android e Windows utilizando a linguagem .NET) ou Ionic (*Framework* para desenvolvimento de aplicações *mobile* híbridas que utiliza JavaScript, HTML, CSS, e *TypeScript*).

Funcionalidades disponíveis no SAP podem ser portadas para dispositivos Android utilizando *WebServices*, e não apenas a visualização de dados, que, embora possa ser muito útil, não permite a modificação ou atualização de dados. A possibilidade de se transferir funcionalidades disponíveis no SAP para aparelhos móveis pode tornar a execução de funções rotineiras ainda mais rápida. O SAP é um ERP completo com todas as funcionalidades que uma empresa pode necessitar, e a possibilidade de manusear suas funcionalidades e informações em dispositivos móveis é um elemento que pode se tornar um diferencial importante na tomada de decisões.

REFERÊNCIAS

ABLESON, W. F.; SEN, R.; KING, C.; ORTIZ, C. E.; Android em ação. Tradução de Eduardo Kraszczuk e Edson Furmankiewicz, Rio de Janeiro: Elsevier, 2012.

ALMEIDA, M. B.; Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares. Ci. Inf., Brasília, v. 31, n. 2, p. 5-13, aug. 2002. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652002000200001&lng=en&nrm=iso. Acesso em: 13 aug. 2019.

DAVENPORT, T. H.; Missão Critica: obtendo vantagem competitiva com os sistemas de gestão empresarial. Porto Alegre: Bookman, 2002.

ELPAIS. Android já é o sistema operacional mais usado do mundo, 04 abr. 2017. Disponível em: https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467_396232.html. Acesso em: 19 mai. 2019.

HIRAMA, K.; FUGITA, H. S.; Soa: Modelagem, Análise e Design, Rio de Janeiro: Elsevier, 2012.

JAMEROV, D.; ISAKOVA, S; *Kotlin in action. Shelter Island*, NY: Manning Publications Co.,set. 2017.

PEREIRA, L. C. O.; SILVA, M. L.; ANDROID PARA DESENVOLVEDORES, RJ: BRASPORT Livros e Multimídia Ltda., 2009.

SAMPAIO, C. SOA e *WebServices* em Java. Rio de Janeiro: Brasport, 2006.

SANTOS, A. A.; KALDEICH, C; SILVA, L. G. C.; Sistemas ERP: Um enfoque sobre a utilização do SAP R/3 em contabilidade e custos; Disponível em: http://www.abepro.org.br/biblioteca/enegep2003_tr0905_0971.pdf. Acesso em 01 jun. 2019.

SAP. ABAP *Web Services*, Disponível em: https://help.sap.com/saphelp_scm700_ehp03/helpdata/en/48/52347a08e672d0e10000000a42189c/frameset.htm. Acesso em 07 abr. 2019.