<http://wiki.freepascal.org/Dialog_Examples>

Application.MessageBox

**25NOV**

Em vez de usar a ShowMessage ou MessageDlg, é bom usar a função Application.MessageBox. Ela utiliza a API do Windows, ficando mais próxima das janelas que a maioria dos programas usam, ao mesmo tempo que já passa o handle proprietário adequadamente, por ser da Application.

Chame assim:

Application.MessageBox('Texto da mensagem', 'Título', mb\_iconinformation + mb\_ok);

Uma vantagem de usar ela é que os botões aparecerão no idioma do Windows, já que é o Windows que cria a janela. Ao usar a MessageDlg o Delphi usa sua VCL, mantendo os botões em inglês.

Ícone da janela de pergunta

Utilize um desses valores para o ícone:

* MB\_ICONINFORMATION – ícone “i”, para informações gerais.
* MB\_ICONERROR – ícone de erro, use para alertas graves ou falhas tratadas.
* MB\_ICONWARNING – ícone de aviso, exclamação geralmente num triângulo amarelo.
* MB\_ICONQUESTION – ícone de pergunta, interrogação.

Exemplo:

if Application.MessageBox('Deseja salvar o texto atual?', 'Salvar', mb\_iconquestion + mb\_yesno) = idYes then

if SaveDialog1.Execute then

Memo1.Lines.SaveToFile(SaveDialog1.Filename);

Botões

Você pode usar constantes definidas na Unit Windows. Veja:

* MB\_OK – apenas o botão de OK, use para notificações.
* MB\_OKCANCEL – botões OK e Cancelar.
* MB\_YESNO – botões Sim e Não.
* MB\_YESNOCANCEL – botões Sim, Não e Cancelar.
* MB\_RETRYCANCEL – botões Repetir e Cancelar.
* MB\_ABORTRETRYIGNORE – botões Abortar Anular, Repetir e Ignorar.

Retorno

A função MessageBox retornará:

* IDOK – se o usuário clicar no OK.
* IDCANCEL – se o usuário clicar no Cancelar.
* IDYES – se o usuário clicar no Sim.
* IDNO – se o usuário clicar no Não.
* IDABORT – se o usuário clicar no Abortar.
* IDRETRY – se o usuário clicar no Repetir.
* IDIGNORE – se o usuário clicar no Ignorar.

Botão Padrão

O botão padrão já vem com o foco, e será acionado ao teclar Enter. Dependendo da situação pode ser desejável que outro botão tenha o foco, sem ser o primeiro, para evitar alguma ação catastrófica desastrosa se o usuário teclar enter sem ler a mensagem. Adicione nas flags (com +) MB\_DEFBUTTON2 para que o botão padrão seja o segundo. Use MB\_DEFBUTTON3 para que seja o terceiro, e MB\_DEFBUTTON4 para que seja o quarto.

Exemplo:

if Application.MessageBox('Tem certeza que deseja sair sem gravar as opções?', 'Aviso', mb\_yesno + mb\_defbutton2) = idYes then

Application.Terminate;

Java

<https://imasters.com.br/linguagens/java/listagem-de-frameworks-java/?trace=1519021197&source=single>

Dificilmente uma corporação terá poder intelectual, financeiro, mão de obra disponível e tempo para competir com a comunidade mundial. Em outras palavras, o arquiteto de software de uma corporação deve conhecer o máximo possível das opções de componentes e frameworks existentes no mercado para não cair no velho e já conhecido buraco de tentar “reinventar a roda” a sua maneira.

Esse é um dos vários tópicos discutido no nosso treinamento de [arquiteto de software Java](http://fernandofranzini.wordpress.com/for-j/). Dentro desse assunto, gostaria de publicar minha lista de frameworks e componentes Java utilizado nas minhas decisões arquiteturais no momento de arquiteturar uma solução. Qualquer sugestão é bem vinda!

## ****Container IoC****

* Spring – <http://www.springsource.org/>
* Guice – <http://picocontainer.codehaus.org/>
* PicoContainer – <http://code.google.com/p/google-guice/>
* EJB – <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>
* CDI – <http://www.oracle.com/technetwork/articles/java/cdi-javaee-bien-225152.html>

## ****Persistência****

* Hibernate – <http://www.hibernate.org/>
* Spring Data – <http://www.springsource.org/>
* iBatis – <http://www.mybatis.org/>
* Torque – <http://db.apache.org/torque/>
* Castor – <http://www.castor.org/>
* Cayenne – <http://cayenne.apache.org/>

## ****Gerenciador de transação****

* Spring AOP transaction – <http://www.springsource.org/>
* EJB (JTA) – <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

## ****Agendamento de tarefas****

* Timer Task JSE – <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/threads/timer.html>
* Quarts – <http://quartz-scheduler.org/>
* EJB Timer Service – <http://docs.oracle.com/javaee/6/tutorial/doc/bnboy.html>

## ****Validação****

* Commons Validator – <http://commons.apache.org/validator/>
* Java Bean Validator – <http://docs.oracle.com/javaee/6/tutorial/doc/gircz.html>

## ****Chamadas assíncronas****

* JSE Executor Services – <http://docs.oracle.com/javase/tutorial/essential/concurrency/exinter.html>
* EJB Asynchronous Method Invocation – <http://docs.oracle.com/javaee/6/tutorial/doc/gkkqg.html>
* EJB Message-Driven Bean – <http://docs.oracle.com/javaee/6/tutorial/doc/gipko.html>

## ****Kit de interface gráficas desktop****

* Swing – <http://docs.oracle.com/javase/tutorial/uiswing/>
* SwingX – <http://swingx.java.net/>
* JGoodies – <http://www.jgoodies.com/>
* SWT – <http://www.eclipse.org/swt/>
* SWT OPAL – <http://code.google.com/a/eclipselabs.org/p/opal/>

## ****RAD interface gráficas desktop****

* Eclipse Windows Builder – <http://www.eclipse.org/windowbuilder/>
* NetBeans Matisse – <http://netbeans.org/features/java/swing.html>

## ****Framework web****

* Click – <http://click.apache.org/>
* Wicket – <http://wicket.apache.org/>
* Spring MVC – <http://www.springsource.org/>
* Struts – <http://struts.apache.org/>
* Mentawai –<http://www.mentaframework.org/>
* JSF – <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>
* VRaptor – <http://vraptor.caelum.com.br/>

## ****Framework web RIA****

* GWT – <https://developers.google.com/web-toolkit/>
* Vaadin – <https://vaadin.com/home>
* ZK Framework – [http://www.zkoss.org](https://vaadin.com/home)/

## ****Framework web full stac****k

* Grails – <http://grails.org/>
* Ruby On Rails – <http://rubyonrails.org/>
* OpenXava – <http://www.openxava.org/>

## ****Provedores de JSF****

* Mojorra – <http://javaserverfaces.java.net/>
* MyFaces – <http://myfaces.apache.org/>
* ADF – <http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html>

## ****Componentes JSF****

* RichFaces – <http://www.jboss.org/richfaces>
* PrimeFaces – <http://primefaces.org/>
* OpenFaces – <http://openfaces.org/>
* IceFaces – <http://www.icesoft.org/>
* EasyFaces – [http://www.easyfaces.com.br](http://www.easyfaces.com.br/)
* Gmaps4jsf – <http://code.google.com/p/gmaps4jsf/>

## ****Extensões JSF****

* PrettyFaces – <http://ocpsoft.org/prettyfaces/>
* Omnifaces – <https://showcase-omnifaces.rhcloud.com/>

## ****Soluções web mobil****

* JSF RichFaces Mobile – <http://www.jboss.org/richfaces>
* JSF PrimeFaces Mobile – <http://primefaces.org/>
* JSF OpenFaces Mobile – <http://openfaces.org/>
* JSF IceFaces Mobile – <http://www.icesoft.org/>
* Spring Mobile – <http://www.springsource.org/>
* JQuery Mobile – <http://jquery.com/>
* DOJO Mobile – <http://dojotoolkit.org/>

## ****Autenticação e autorização web****

* Spring Security – <http://www.springsource.org/>
* JGuard – <http://jguard.xwiki.com/xwiki/bin/view/Main/WebHome>
* JEE Container Managed Security (Configuração especifica para cada provedor de container JEE). Veja o Tomcat – <http://tomcat.apache.org/tomcat-7.0-doc/realm-howto.html>

## ****JEE web container standalone****

* Tomcat – <http://tomcat.apache.org/>
* Jetty – <http://jetty.codehaus.org/jetty/>
* JBoss – <http://www.jboss.org/>

## ****JEE EJB container (web profile e JEE Full)****

* Tomee – <http://openejb.apache.org/apache-tomee.html>
* Simpas – <http://siwpas.mechsoft.com.tr/>
* Resin – <http://www.caucho.com/resin-application-server/>
* JBoss – <http://www.jboss.org/>
* Glassfish – <http://glassfish.java.net/>
* Webshere – <http://www-142.ibm.com/software/products/br/pt/appserv-was>
* WebLogic – <http://www.oracle.com/us/products/middleware/application-server/weblogic-suite/index.html?ssSourceSiteId=ocombr>
* Geronimo – <http://geronimo.apache.org/>

## ****Relatórios****

* iReport – <http://jasperforge.org/projects/ireport>
* DynamicReport – [http://dynamicreports.sourceforge.net](http://dynamicreports.sourceforge.net/)
* Birt – <http://www.eclipse.org/birt/phoenix/>
* iText – <http://itextpdf.com/>
* PDFBox – [http://pdfbox.apache.org/](http://itextpdf.com/)

## ****Gerador de gráficos charts****

* JFreechart – <http://www.jfree.org/jfreechart/>
* HighCharts – <http://www.highcharts.com/>

## ****Web Services REST****

* Jersey – <http://jersey.java.net/>
* Restlet – <http://www.restlet.org/>
* RestEasy – <http://www.jboss.org/resteasy/>
* Serfj – <http://serfj.sourceforge.net/>
* CXF – <http://cxf.apache.org/>

## ****Gerenciador de e-mails****

* Commons Email – <http://commons.apache.org/email/>
* JavaMail – <http://www.oracle.com/technetwork/java/javamail/index.html>

## ****Web Services SOAP****

* Metro – <http://jax-ws.java.net/>
* Axis – <http://axis.apache.org/axis/>
* XFire – <http://xfire.codehaus.org/>
* CXF – [http://cxf.apache.org/](http://xfire.codehaus.org/)

## ****JavaScript****

* DOJO Toolkit – <http://dojotoolkit.org/>
* JQuery – <http://jquery.com/>
* Ext-JS – <http://docs.sencha.com/ext-js/4-0/>
* DWR – <http://directwebremoting.org/dwr/index.html>

## ****Integração com redes sociais****

* Spring Social – <http://www.springsource.org/>

## ****Cache****

* Space4J – <http://www.space4j.org/>
* Memcache – <http://memcached.org/>
* Ecache – <http://ehcache.org/>
* Prevayler – <http://prevayler.org/>
* JBoss Infinispan – <http://www.jboss.org/infinispan.html>

## ****NoSQL****

* MongoDB – <http://www.mongodb.org/>
* Neo4j – <http://neo4j.org/>
* Db4o – <http://www.db4o.com/>
* MemcachedDB – <http://memcachedb.org/>
* CouchDB – <http://couchdb.apache.org/>
* Project Voldemort – <http://project-voldemort.com/>
* MarkLkogic Server – <http://www.marklogic.com/>
* BaseX – <http://basex.org/>
* SimpleDB – <http://aws.amazon.com/simpledb/>
* HBase – <http://hbase.apache.org/>
* Casandra – <http://cassandra.apache.org/>
* Hypertable – <http://hypertable.org/>
* OrientBD – <http://www.orientechnologies.com/>

## ****SGDB Java embutidos****

* HyperSQL – <http://hsqldb.org/>
* JavaDB – <http://www.oracle.com/technetwork/java/javadb/overview/index.html>
* Apache Derby – <http://db.apache.org/derby/>
* H2Database – [http://www.h2database.com](http://www.h2database.com/)

## ****Testes****

* DBUnit – <http://www.dbunit.org/>
* JUnit – <http://www.junit.org/>
* EJBUnit – <http://ejb3unit.sourceforge.net/>
* JSFUnit – <http://www.jboss.org/jsfunit/>

## ****Loggin****

* jse – <http://docs.oracle.com/javase/7/docs/api/java/util/logging/package-summary.html>
* Log4J – <http://logging.apache.org/index.html>

## ****Testes cargas e perfomance****

* JMeter – <http://jmeter.apache.org/>
* WebLoad – <http://www.webload.org/>

## ****JSON****

* JSON – <http://www.json.org/java/>
* XStream – <http://xstream.codehaus.org/json-tutorial.html>
* Google-JSON – <http://code.google.com/p/google-gson/>

## ****Boletos bancário****

Boletos Bancário – [http://www.boletobancario.com.br](http://www.boletobancario.com.br/)

Bopepo – <http://www.jrimum.org/bopepo/>

Stella – <http://github.com/caelum/caelum-stella/wiki/Gerando-boleto>

## ****Manipulaçao de arquivos MS Office****

Apache POI – <http://poi.apache.org/>

JExcelApi – <http://jexcelapi.sourceforge.net/>

Docx4j – <http://dev.plutext.org/trac/docx4j>

Java Docx – <http://www.javadocx.com/>

Aspose – <http://www.aspose.com/categories/java-components/aspose.total-for-java/default.aspx>

## ****Gerais****

Maven – Gerenciador de dependências – <http://maven.apache.org/>

JLicense – Gerenciador de licenciamento de uso de software – <http://www.websina.com/products/jlicense.html>