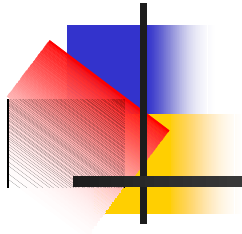


# Implementing applications using Firebird API in Delphi, Lazarus and Free Pascal



Luiz Paulo de Oliveira Santos

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# Case in discussion

---

This presentation is based in  
Brasilians cases. Some diferences can  
be founf in other countries. Since  
language until the programming  
tecniques.

A decorative graphic on the left side of the title, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# How API works - requirements

---

## n Application Programming Interface

The applications need be concepted for use API. Need be designed for this.

A decorative graphic consisting of overlapping yellow, red, and blue squares with a black crosshair.

# Advantages

---

- n Principal Advantages
  - n Final code smaller
  - n Code clear
  - n Total control in application
  - n Works in diferent languages

A decorative graphic consisting of overlapping yellow, red, and blue squares with a black crosshair.

# Problems

---

- n Principal problems
  - n Difícult to depure
  - n Difícult to implement
    - n There is a lot of easy and good components

A decorative graphic consisting of overlapping red, yellow, and blue squares with a black crosshair.

# Security Problem

---

## n Password to users

Special care need be implement to protect the password and the user of the database for API applications.

isc\_user, isc\_database, isc\_password

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# How API works - requirements

---

## n Language

The choose of Language is very important. Because some languages have restrictions for API.

Delphi, Lazarus and Free Pascal

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# How API works - requirements

---

## n Operating System

Diferents operation system can reduce the portability of code and the careful of the programmer.



A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares with a black crosshair.

# How API works - requirements

---

## n Dinamic Projects

Projects very dynamics can have problems with API.

Good project for good work.

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# How API works - requirements

---

## n Error Handling

One of the principal problems:

- error handling for applications based in API.

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# How API works - requirements

---

## n Basic Requeriments

Some API applications need special rights in server.

A decorative graphic consisting of overlapping colored squares (yellow, red, blue) and a black crosshair.

# Procedures

---

## n Basic Requeriments

Determine the plataform.

Coding the application

Compile and link

Test the application

Deploy to Clients

A decorative graphic consisting of overlapping red, yellow, and blue squares and a black crosshair.

# Procedures

---

n About API:

The calls are C programming.

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and rectangles, with a black crosshair-like structure.

# Principals functions - Connect

---

<i>isc_attach_database()</i>	Connects to a database and establishes initial parameters for database access, such as number of cache buffers to use; uses a previously declared and populated DPB
<i>isc_database_info()</i>	Retrieves requested information about an attached database, such as the version of the on-disk structure (ODS) that it uses
<i>isc_detach_database()</i>	Disconnects from an attached database and frees system resources allocated to that attachment
<i>isc_drop_database()</i>	Deletes a database and any support files, such as shadow files

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares with a black crosshair.

# Principals functions - Errors

---

*isc\_interprete()*

Capture InterBase error messages to a buffer

*isc\_print\_sqlerror()*

Display an SQL error message

*isc\_print\_status()*

Display InterBase error messages

*isc\_sqlcode()*

Set the value of SQLCODE

*isc\_sql\_interprete()*

Capture an SQL error message to a buffer

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares with a black crosshair.

# Principals functions - Events

---

<i>isc_event_block()</i>	Allocate event parameter buffers
<i>isc_wait_for_event()</i>	Wait for a synchronous event to be posted
<i>isc_que_events()</i>	Set up an asynchronous event and return to application processing
<i>isc_event_counts()</i>	Determine the change in values of event counters in the event parameter buffer
<i>isc_cancel_events()</i>	Cancel interest in an event



A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares with a black crosshair.

# Principals functions - Events

---

<i>isc_event_block()</i>	Allocate event parameter buffers
<i>isc_wait_for_event()</i>	Wait for a synchronous event to be posted
<i>isc_que_events()</i>	Set up an asynchronous event and return to application processing
<i>isc_event_counts()</i>	Determine the change in values of event counters in the event parameter buffer
<i>isc_cancel_events()</i>	Cancel interest in an event

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares with a black crosshair.

# Important know about API

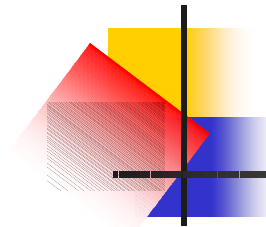
---

How work with Services

Possibility of manage Limbo Transactions

Access Security Database

C values to calls

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# An example of application

---

- n Brazilian SIAGES

- n Sistema Aberto de GESTão empresarial

- n Opened system for enterprise management

Source code in:

[http://codigolivre.org.br/frs/?group\\_id=1519](http://codigolivre.org.br/frs/?group_id=1519)

# Screen shoot of SIAGES

**Cientes**

Cadastro Pesquisa

Incluir Excluir Gravar Cancelar Ajuda Sair

Código 1590 ☒ Consumidor Final ☐ Inclusão Persistente

Nome Douglas Fernando Scheibler

Fantasia Douglas Fernando Scheibler

Endereço Rua Leopoldo Schneider,1466

Bairro Canabarro

Cidade Teutônia UF RS

CEP 95890-000 Caixa Postal

Fone/Fax 3762-9497

Contato Raquel Wessel Scheibler

Imagem

e-mail douglas@solis.coop.br

CGC Inscrição Estadual

Praça 2 TEUTONIA

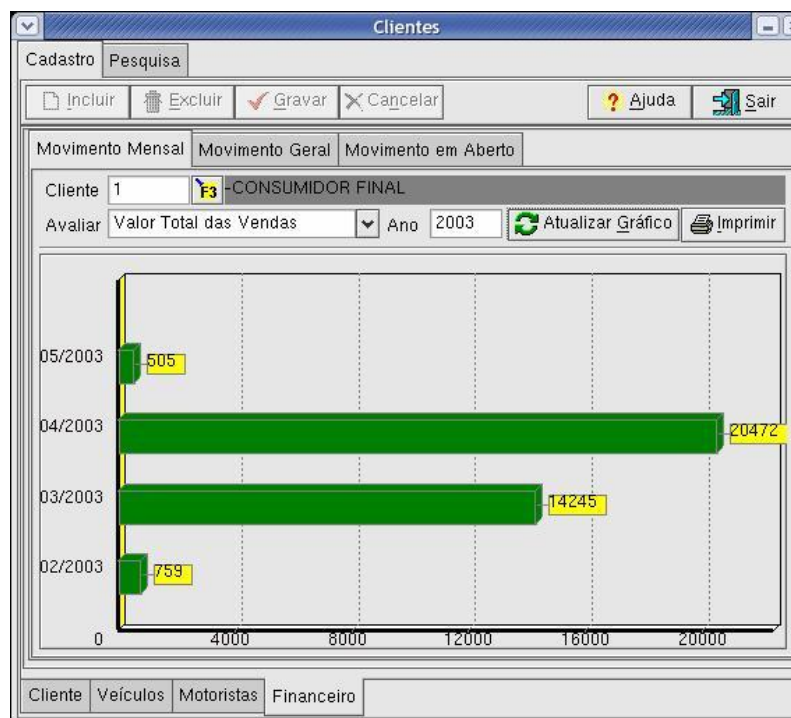
Situação 1 Cliente Liberado

Classific. 1 Classe A

Tipo Ativid. 1 DESCONHECIDO

Vendedor 1 VENDEDOR

Cliente Veículos Motoristas Financeiro



A decorative graphic consisting of overlapping colored squares (yellow, red, blue) and a black crosshair.

# What is new in version 2.0

---

## **Extended `isc_dsql_info()` API call**

D. Yemanov

The function call `isc_dsql_info()` has been extended to enable relation aliases to be retrieved, if required.

---

## **API identifies client version**

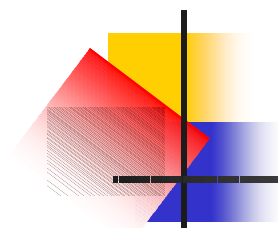
N. Samofatov

C/C++ client interface version `FB_API_VER` is defined as 20 for Firebird 2.0 in `ibase.h`. More information to come.

---



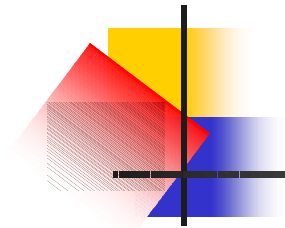
Prague 2005

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# Example in Delphi

---

## Run Borland Delphi

A decorative graphic consisting of overlapping colored squares (red, yellow, blue) and a black crosshair.

# Example in Free Pascal

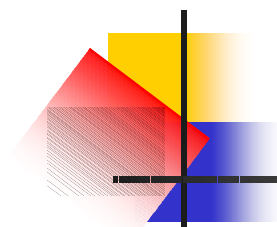
---

## Run Free Pascal





Prague 2005

A decorative graphic on the left side of the slide, consisting of overlapping red, yellow, and blue squares and a black crosshair.

# Example in Lazarus

---

## Run Lazarus





Prague 2005



# The End...

I wait be contributed with the viewers  
and some questions please write to:

PRAGA@JOBVOX.COM.BR