

BLAISE PASCAL MAGAZINE 118/119



Multiplatform / Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js
Databases / CSS Styles / Progressive WebApps
Android / IOS / Mac Windows & Linux



Lição nº 1 de programação com Pascal para crianças

Gamificação: leve as crianças a sério

A filosofia por trás dessa nova abordagem.

Experiências com o Lazarus no Raspberry Pi

Instalação do Lazarus para Raspberry 2/3/4/5/Zero no Linux
integração do Lazarus com todos os tipos de sistemas operacionais

Boas notícias da categoria: PFAS removido

Novo material de filtragem pode remover produtos químicos de longa duração da água

O novo Delphi UPDATE 12.2

Componentes para desenvolvedores (kbmMW)

Criação de programas com o C4D para iniciantes

Visualização do Lazarus TRUNK VERSÃO 4.0

Integração de IA / Navegador de arquivos / IDE Scout / Pesquisa instantânea / Minimapa
Escrevendo código Delphi com o Chatgpt 4

O jogo de quebra-cabeça Tiles, de David Dirkse

Fastreport - atualização dos produtos Delphi e Lazarus para a versão 2024.2

Pré-visualização do AI AND Lazarus / FPC

FMX para Lazarus

Como criar um aplicativo da web FMX no FPC / Lazarus



CONTEÚDO

ARTIGOS

<i>De seu editor</i>	Página 4
<i>De nosso consultor técnico, Jerry King</i>	Página 5
Lição nº 1 de programação com Pascal para crianças	Página 7
Gamificação: leve as crianças a sério	Página 23
<i>A filosofia por trás dessa nova abordagem.</i>	
Experiências com o Lazarus no Raspberry Pi	Página 30
Instalação do Lazarus para Raspberry 2/3/4/5/Zero no Linux	
<i>integração do Lazarus com todos os tipos de sistemas operacionais</i>	
Boas notícias da categoria: PFAS removido	Página 34
<i>novo material de filtragem pode remover produtos químicos de longa duração da água</i>	
O novo Delphi UPDATE 12.2	Página 40
Componentes para desenvolvedores (kbnMW)	Página 46
<i>Criação de programas com o C4D para iniciantes</i>	
Visualização do Lazarus TRUNK VERSÃO 4.0	Página 50
<i>Integração de IA / Navegador de arquivos / IDE Scout / Pesquisa instantânea / Minimapa</i>	
Escrevendo código Delphi com o Chatgpt 4	Página 57
O jogo de quebra-cabeça Tiles, de David Dirkse	Página 62
Fastreport - atualização dos produtos Delphi e Lazarus para a versão 2024.2	Página 70
Pré-visualização do AI AND Lazarus / FPC	Página 76
FMX para Lazarus	Página 82
<i>Como criar um aplicativo da web FMX no FPC / Lazarus</i>	

PUBLICIDADE

Components4Developers	Página 108
David Dirkse Book Computer / Graphs / Games & Math	Página 33
Database Workbench / Upscene	Página 45
Fastreport UPDATES	Página 6
LIBRARY Stick including USB Card	Página 22
Lazarus HANDBOOK	Page 29
Lazarus HANDBOOK + SUBSCRIPTION	Página 69
SUPERPACK	Página 61
Ukraine Special Offer	Página 107



Pascal é uma linguagem de programação imperativa e procedural, projetada por Niklaus Wirth (esquerda abaixo) em 1968-69 e publicada em 1970, como uma linguagem pequena e eficiente destinada a incentivar boas práticas de práticas de programação usando programação estruturada e estruturação de dados. Um derivado conhecido como Object Pascal, projetada para programação orientada a objetos, foi desenvolvida em 1985. O nome da linguagem foi escolhido para homenagear o matemático, inventor da primeira calculadora: Blaise Pascal (veja o canto superior direito).

Editora: PRO PASCAL FOUNDATION em colaboração
© Foundation Supporting Programming Language Pascal



CONTRIBUINTES

Stephen Ball http://delphiaball.co.uk DelphiABall	Dmitry Boyarintsev dmitry.living @ gmail.com	Michaël Van Canneyt ,michael @ freepascal.org	Marco Cantù www.marcocantu.com marco.cantu @ gmail.com
David Dirkse www.davdata.nl mail: David @ davdata.nl	Benno Evers b.evers @ everscustomtechnology.nl	Bruno Fierens www.tmssoftware.com bruno.fierens @ tmssoftware.com	Holger Flick holger @ flixments.com
Mattias Gärtnernc- gaertnma@netcologne.de	Max Kleiner www.softwareschule.ch max @ kleiner.com	John Kuiper john_kuiper @ kpnmail.nl	Wagner R. Landgraf wagner @ tmssoftware.com
Vsevolod Leonov vsevolod.leonov@mail.ru	Andrea Magni www.andreamagni.eu andrea. magni @ gmail.com www.andreamagni.eu/wp		Helmut Elsner Korrektor der Deutschen Ausgabe helmut.elsner@live.com
		Paul Nauta PLM Solution Architect CyberNautics paul.nauta @ cybernautics.nl	
Kim Madsen www.component4developers.com kbmMW		Boian Mitov mitov @ mitov.com	
	Jeremy North jeremy.north @ gmail.com	Detlef Overbeek - Editor in Chief www.blaisepascal.eu editor @ blaisepascal.eu	
Anton Vogelaar ajv @ vogelaar-electronics.com	Danny Wind dwind @ delphicompany.nl	Jos Wegman Corrector / Analyst	Siegfried Zuhr siegfried @ zuhr.nl

Editor-chefe

Detlef D. Overbeek, Holanda Tel: Celular: +31 (0)6 21.23.62.68

Notícias e comunicados à imprensa somente por e-mail para editor@blaisepascal.eu

As assinaturas podem ser feitas on-line em www.blaisepascal.eu ou por ordem escrita, ou enviando um e-mail para office@blaisepascal.eu.
As assinaturas podem ser iniciadas em qualquer data. Todas as edições publicadas no ano civil da assinatura também serão enviadas.
As assinaturas têm duração de 365 dias. As assinaturas não serão prorrogadas sem aviso prévio. O recibo de pagamento será enviado por e-mail.
As assinaturas podem ser pagas enviando o pagamento para: ABN AMRO Bank Account no. 44 19 60 863 ou por cartão de crédito ou PayPal
Nome: Pro Pascal Foundation (Stichting Ondersteuning Programeertaal Pascal) IBAN: NL82 ABNA 0441960863 BICABNANL2A VAT/NL814254147B01
Departamento de Assinaturas Edelstenenbaan 21 / 3402 XA IJsselstein, Holanda + 31 (0) 6 21.23.62.68 office@blaisepascal.eu

Marcas registradas Todas as marcas registradas usadas são reconhecidas como propriedade de seus respectivos proprietários. Advertência Embora nos esforcemos para garantir que o que é publicado na revista esteja correto, não podemos aceitar a responsabilidade por quaisquer erros ou omissões.
Se o senhor notar algo que possa estar incorreto, entre em contato com o Editor e publicaremos uma e publicaremos uma correção, se for o caso.

Member of the Royal Dutch Library

KB Koninklijke Bibliotheek
BIBLIOTÉCA REAL

Membro e doador da



WIKIPEDIA

SUBSCRIPTIONS 2024 prices)

Electronic Download Issue (8 per year) ±60 pages : **R\$ 250**

AVISO DE DIREITOS AUTORAIS

Todo o material publicado na Blaise Pascal é protegido por direitos autorais

© SOPP Stichting Ondersteuning Programeertaal Pascal, a menos que seja não pode ser copiado, distribuído ou republicado sem permissão por escrito. Os autores concordam que o código associado a seus artigos será disponibilizado aos assinantes após a publicação, colocando-o no site do PGG para download, e que os artigos e o código serão colocados em mídia de armazenamento de dados distributiva. O uso das listas de programas pelos assinantes para fins de pesquisa e estudo é permitido, mas não para fins comerciais. O uso comercial de O uso comercial de listagens de programas e códigos é proibido sem a permissão por escrito do autor.



De seu editor

Olá a todos vocês,

Essa questão levou muito tempo porque alguns itens tiveram que ser escritos quando o desenvolvimento do item foi finalizado.

Tudo é feito no limite

Fizemos tantas mudanças drásticas no FreePascal/**Lazarus** que é muito difícil de acreditar ...

Como está escrito na capa: o futuro está chegando....

Estes são os principais itens

- Assinatura de código para o instalador do **Lazarus**/FPC - uma irritação de longa data que precisava ser resolvida
- Integração de IA / Navegador de arquivos / IDE Scout / Pesquisa instantânea / Minimapa
- Criamos até mesmo um editor de pascal para crianças, para que qualquer pessoa possa começar a aprender pascal de uma maneira muito simples: brincando.

Cada um deles é um enorme passo à frente e esses novos itens serão mostrados pela primeira vez no Encontro de Outubro em Colônia.

Nesta edição, você encontrará muitos desses itens.

A maioria delas será explicada de forma mais explícita em uma próxima edição.

Eles são tão novos que precisamos documentá-los mais detalhadamente.

Há um item que quero resolver o mais rápido possível e já fizemos algum progresso:

O tema escuro para o **Lazarus** no Windows/Linux e Mac.

Estamos trabalhando duro para isso. Ele será resolvido primeiro como uma explicação passo a passo e, depois disso, criaremos um componente que possa ser definido, para que seja fácil mudar.

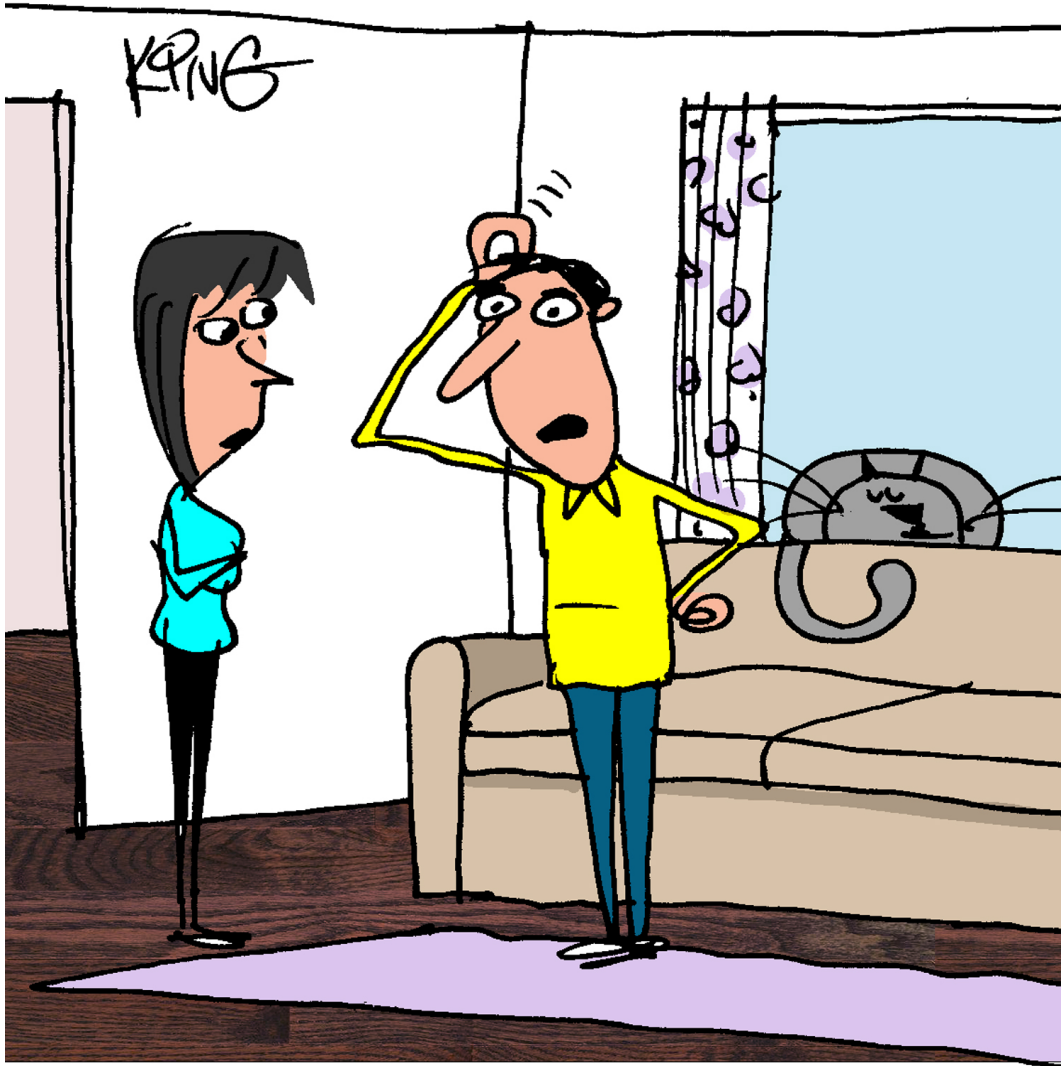
A maioria desses itens já pode ser encontrada na versão trunk.

Por enquanto - estou muito ocupado - não digo mais nada. Boa leitura.

Detlef



de nosso consultor técnico, Jerry King



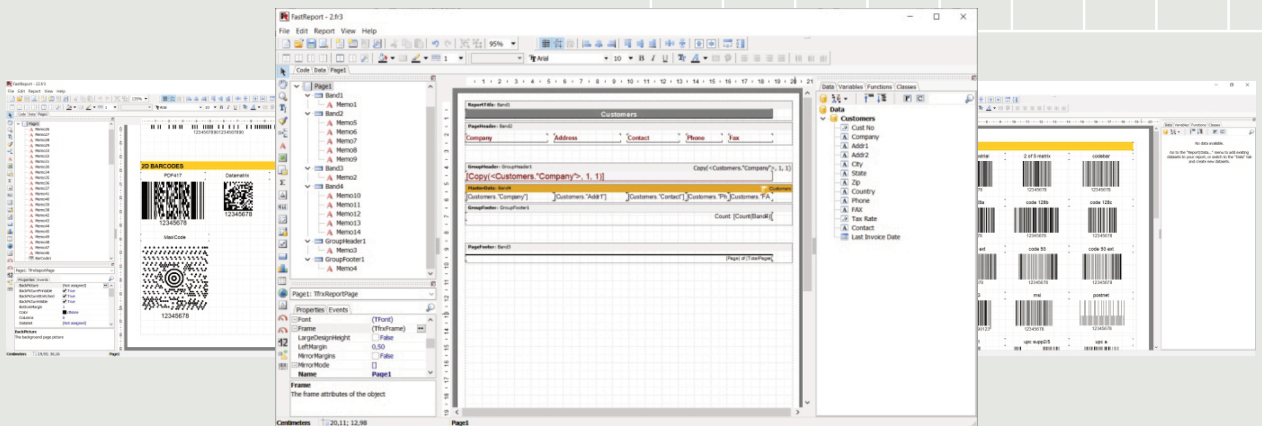
Acho que esqueci de carregar o telefone, o computador, a casa, o Apple Watch, o Ipad, a caneta inteligente, o Fitbit, os Air Pods e o mouse.





Create professionally designed reports with minimal effort with **FastReport VCL**.

Create professionally designed reports with minimal effort with **FastReport VCL**.



Multiple graphical elements for information visualization, export filters to 30+ formats, easy integration with data, secure storage. **FastReport VCL**, with its simplicity, convenience and small distribution size, is able to provide proper functionality and speed on any modern computer.

In the latest version 2024.2:

A new package with visual components TfrTreeView. Presentation of data in an intuitive way.

Support for GeoJSON and TopoJSON formats in the FastReport VCL Maps object.

Lazarus support in FastQueryBuilder. Integration into Lazarus projects and data workflow improvements.

Try the demo today to save time and resources on report creation.

INTRODUÇÃO

Até mesmo algumas pessoas mais velhas ou desenvolvedores interessados nos problemas desafiadores para explicar a programação a uma criança. Quando comecei este projeto, tinha em mente torná-lo muito fácil e compreensível e, portanto, às vezes, ser um pouco infantil seria muito bom. Eu estava pensando em uma lição a partir dos 8 anos de idade. Então, depois de pensar e tentar entrar em um estado de espírito filosófico, repensando constantemente, eu sabia o que poderia fazer. Uma criança, assim como muitos adultos, olha para o celular quase que constantemente e quase nunca o deixa de lado.

Vejo isso como uma oportunidade de fazer o que quero:

fazer com que uma criança se interesse pelo material de que é feita a tela de um telefone e ir quase literalmente direto ao ponto:

O PIXEL.

Portanto, vamos nos aprofundar nos fundamentos e no funcionamento do sistema.

O que, por si só, já é muito interessante.

PONTO DE PARTIDA DA LIÇÃO

A tela de um telefone celular fez com que nos comportássemos de forma muito diferente do que costumávamos fazer. Com muito mais contatos via celular, enviando muitas mensagens e imagens.

De uma tela muito simples até uma tela com muitos pixels (pontos), onde é possível até mesmo ver o texto que faz, ajustar e alterar e ou jogar.

(Todo texto em itálico tem o objetivo de explicar e criar exemplos adicionais)

A TELA DE UM TELEFONE

HISTÓRIA DO TELEFONE CELULAR

Para começar, é importante saber que todas as crianças (e adultos) olham para o celular (tela).

Primeiro, explicaremos como funciona a tela de um telefone celular.

Vamos ver algumas telas do passado até algo que provavelmente será lançado no próximo ano.

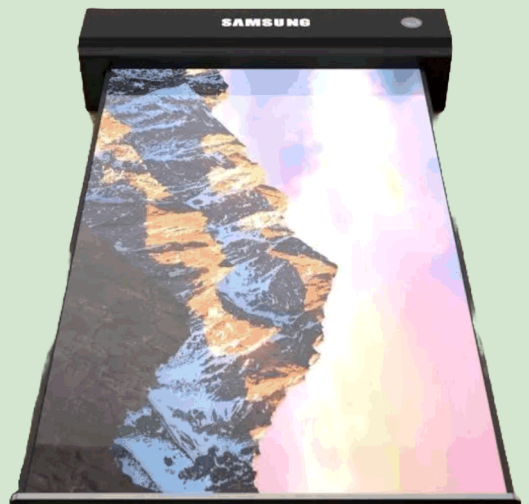
1-Exemplo mostrando um celular de hoje, do futuro e do passado:



O primeiro Philips Carry-able Mobile
Apenas uma tela numérica

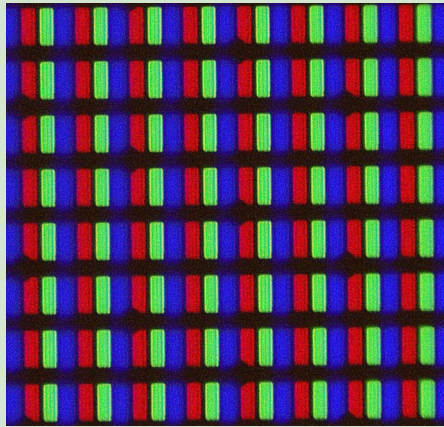


O Nokia:
Sem tela para fotos



O mais novo telefone com rolagem da Samsung Uma vez desenrolado: tão grande quanto um pequeno tablet

2-(Superfície - topo) de forma bastante ampliada.



muitos pixels (pontos), tantos que você pode até exibir uma espécie de imagem tridimensional (quase real).

Essa imagem consiste em pontos, cada um dos quais criado por meio de um programa muito pequeno. Para explicar isso, que é muito interessante, tiramos algumas fotos da tela



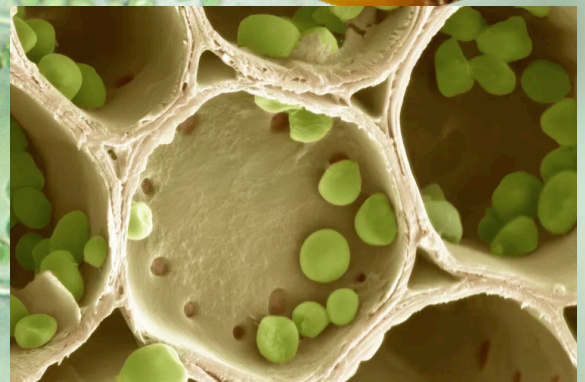
Você pode pensar nisso como uma flor. Primeiro, veja a flor como um todo, depois, com um pouco de ampliação, você vê as pétalas e, cada vez mais de perto, pode até mesmo ver as células individuais, como com uma lupa.

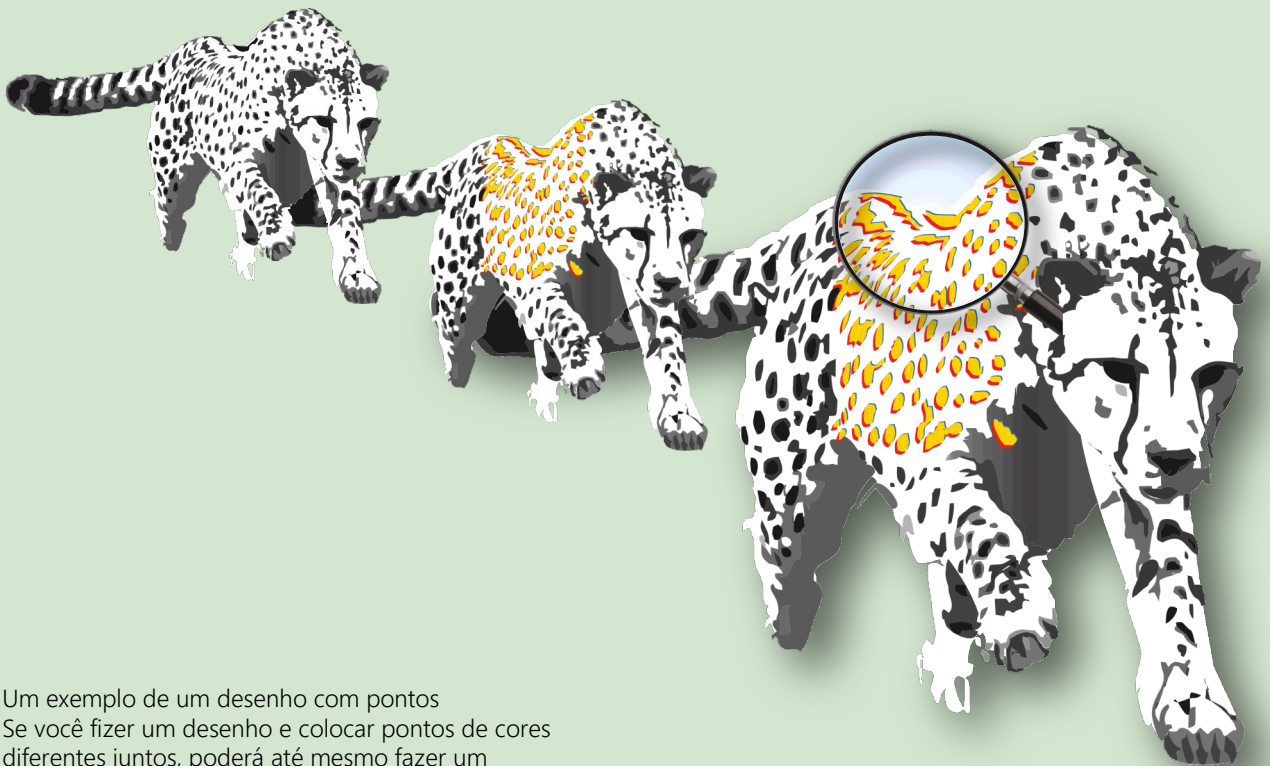
Eventualmente, você pode até olhar DENTRO das células, que novamente têm partes menores.

Com uma tela, não é diferente. Como gostamos de ver cores, agrupamos os pontos com cores diferentes.

O fato de esses pontos serem muito pequenos possibilita a criação de cores com eles. Se você pegar um novo telefone celular, ele terá muitos pontos.

Chamamos isso de pixels, que você pode ver individualmente como um ponto. Todos esses pontos podem trabalhar juntos para mostrar um desenho.





Um exemplo de um desenho com pontos
Se você fizer um desenho e colocar pontos de cores diferentes juntos, poderá até mesmo fazer um desenho colorido.



Funciona da mesma forma que a tela do seu celular. Como o computador (chip) do seu celular pode fazer cálculos super rápidos, você vê uma imagem, em vez de todos os pontos individuais.

Parece que estão todos lá ao mesmo tempo. Isso é muito útil. Vamos usar isso mais tarde.

Se você observar um ponto muito simples (um único), esse ponto tem algumas características especiais muito boas.



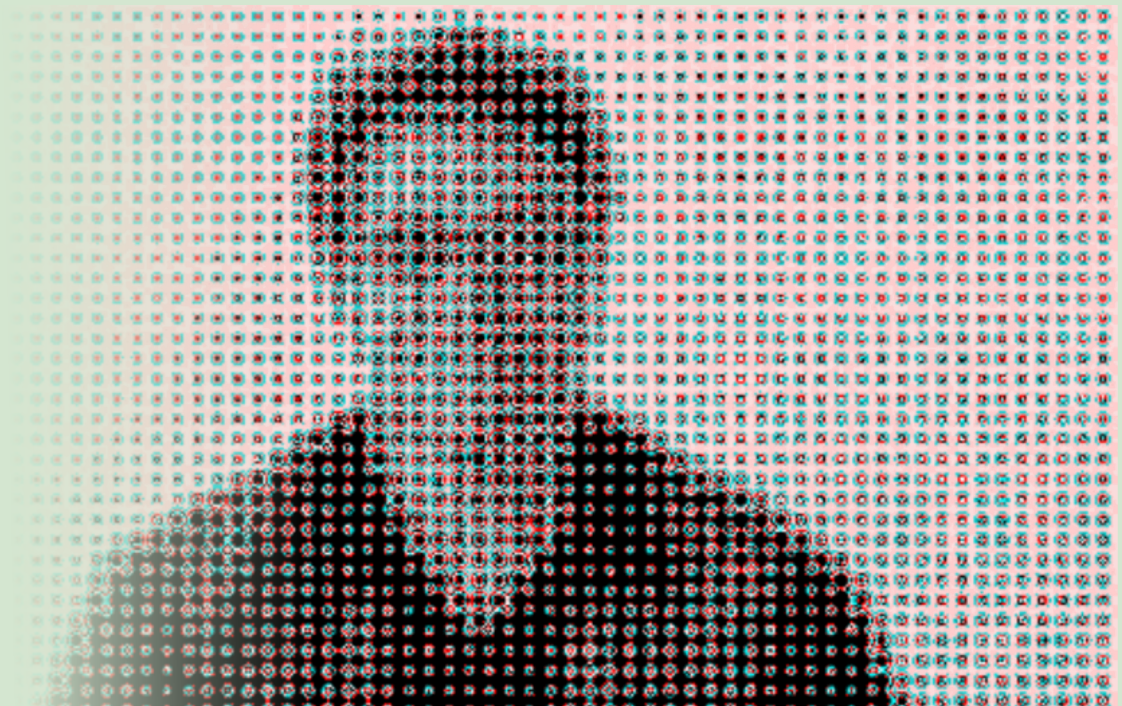
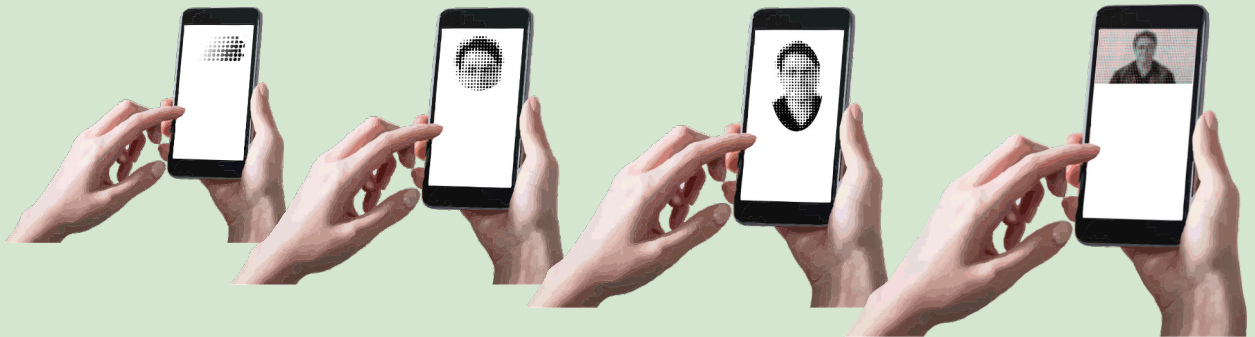
Se você colocar esse ponto em qualquer pedaço de papel, poderá começar com esse único ponto. O que há de tão especial nisso?

Você acabou de criá-lo. Um ponto. Muito simples. Ele é pequeno. Não se move.

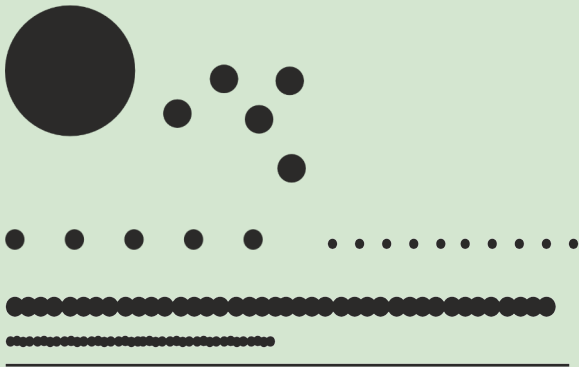
Ele fica em algum lugar e, portanto, tem uma espessura. Um tipo de propriedade. Não é realmente espessa, mas tem algum tipo de espessura. Ou uma largura - bem, muito pequena - mas poderíamos medi-la.

Podemos usar essas propriedades mais tarde. Todas essas coisas que conhecemos em nosso próprio ambiente têm propriedades.

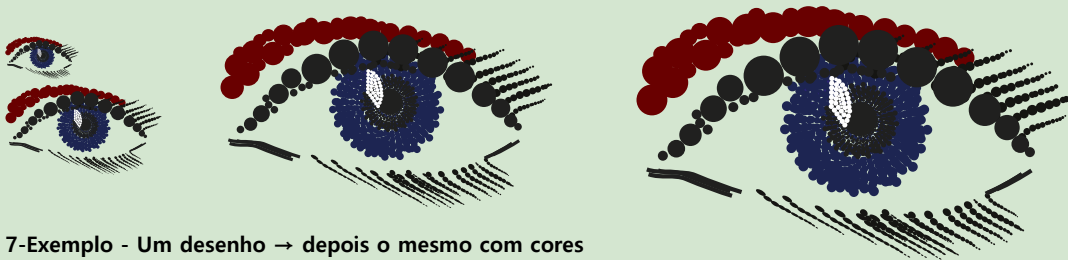
4 - Mostrar retângulo de visualização, mais ou menos como uma tela de celular



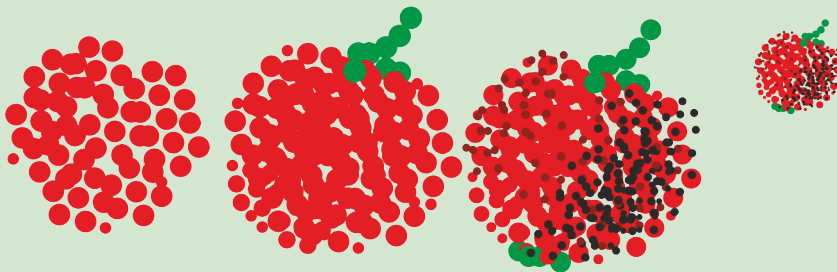
5-Exemplo - Começando com o preto. Primeiro alguns pontos, depois uma linha:



6-Exemplo - Um círculo, depois uma elipse e, por fim, um desenho: (tudo em preto e branco)



7-Exemplo - Um desenho → depois o mesmo com cores



8-Assista ao vídeo aqui.—



Se você quiser ativar o vídeo: clique nele

Se isso não funcionar, aqui está o link <https://youtu.be/m8c1CAT2zEI>



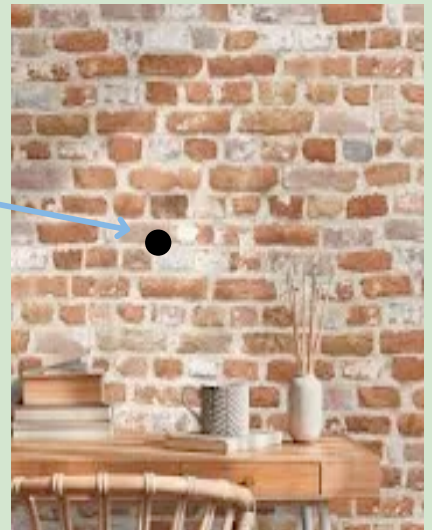
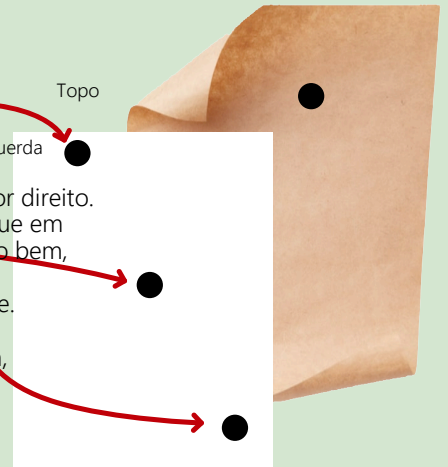
Vamos começar com a primeira lição

Por exemplo, se o ponto estiver em uma folha de papel, ele também terá seu próprio lugar. É claro que pode ser na parede, na praia ou em um jardim. Mas, em uma folha de papel branco, é possível ver claramente. E então você também descobre onde ele está.

Por exemplo, quase no meio, no canto superior esquerdo ou no canto inferior direito. - Não importa onde esteja - Um ponto é um ponto, desde que você o coloque em algum lugar. Você pode fazer isso com um lápis ou um lápis de cor ou, muito bem, com uma caneta-tinteiro e tinta, ou com pincel e tinta. Se você olhar para a ponta, sempre poderá encontrá-la, pois ela não se move.

É por isso que você pode dizer, por exemplo, que está um pouco à esquerda, -ou 5 cm a partir do topo - da borda - no lado esquerdo ou 10 cm da borda. Mas você precisa indicar de que lado está. Criamos algo para isso.

Topo
Esquerda



Se um ponto em uma folha de papel **puder ser encontrado de cima a 5 cm**, então você já tem uma boa pista. Melhor ainda é indicar a distância do **lado esquerdo** ou do **lado direito** da folha de papel.

Então você pode fazer uma cruz bem pequena: x
E esse ponto está sempre no mesmo lugar: **X** ← (ampliado)
No meio dessa cruz está o ponto (vermelho)
e, nessa distância em que se encontra, dissemos que essa é uma propriedade do ponto.

Por exemplo **esquerda 5**, de cima **top 5**.
Em linguagem de computador, chamamos isso de **propriedade**.
Isso significa que, em todas as folhas de papel do mundo, o ponto está exatamente no mesmo lugar.

Isso é conveniente e, para que não nos esqueçamos, demos à **esquerda 5** um nome bem curto: **X(5)**
(5 cm a partir da esquerda)
e para a altura Y(5) - que é a partir do topo da folha de papel - as distâncias que mencionei anteriormente.

X e Y são apenas algumas letras do ALFABETO que escolhemos.



(Mas você poderia ter escolhido qualquer outra letra para a coordenada)
Por exemplo, as letras iniciais de seu nome e do nome de seu amigo.

H = Harry e M de Margaret também teriam funcionado.

Mas no mundo dos computadores, para deixar claro para todos, escolhemos as letras:

Seja na **Índia**, na **Alemanha**, na **Itália** ou em um país da **África**.

Eles sempre usam o mesmo **x** e **y**. Mas não **DEVE** ser assim.

Essas **PROPRIEDADES** são muito úteis e são muito usadas na linguagem de computação.

Se você mover esse ponto, mudaremos a PROPRIEDADE de X de 5 para 15.

A altura permanece a mesma. Não precisamos fazer nada a respeito disso.

→ *Faça uma observação em algum lugar do papel.*

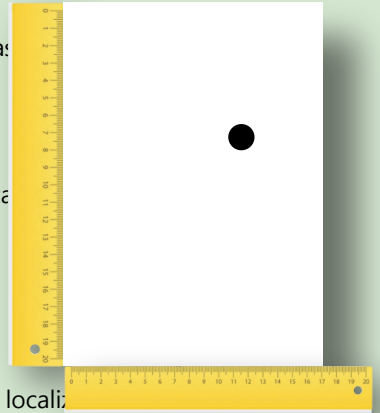
O ponto agora está em algum lugar. Agora podemos medir o ponto.

Descubra onde ele está exatamente no papel.

Se o medirmos com uma régua, você poderá medir que, do lado esquerdo, ele está localizado em algum lugar em um número ... de cm.

Do topo do papel, é claro, ele também está em algum lugar.

Também podemos medir isso. Quantos centímetros são?



CRIAÇÃO DE CÓDIGO - PSEUDOCÓDIGO

Também podemos tentar fazer isso em linguagem de computador.

Isso poderia ter a seguinte aparência:

- *Primeiro, escrevemos em linguagem humana o que queremos fazer -*

- eles chamam isso de código ainda não real (**Pseudo code**)

Isso não é uma doença assustadora, **significa Pseudo**.

É originário de um idioma antigo: a palavra grega **PSEUDES** que significa **não real**.

→ *Faça uma observação em **X (5)** (em centímetros) - lado esquerdo - e **Y (5)** (contando de cima para baixo - a altura).*

Mova o ponto na folha de papel para **X(10)** , **Y (5)** .

→ *Podemos criar esse miniprograma da mesma forma.*

Esse pseudocódigo é uma ótima maneira de escrever suas ideias divertidas para criar um programa real.

É útil ter algum tipo de história sobre o que você quer fazer.

Assim, você se lembrará mais tarde e, além disso, seu namorado ou namorada também entenderá.

Porque ela pode simplesmente ler.

Para entender como dar um comando em nossa linguagem de programação, veja um exemplo.

Na verdade, ele é bem simples.

A linguagem de programação Pascal tem algo muito especial.

Você pode lê-lo.

Sempre começamos com uma palavra como, por exemplo:

```
Begin // → isso é literalmente o começo
// vamos mover o X10 (que é 10 cm) para uma nova posição X15 - que está em 15 cm
// e, em seguida, preenchamos o comando:
X(10) := X(15)
// Esse parêntese (colchete) tem o objetivo de deixar claro que o texto pertence ao mesmo conjunto
// os dois caracteres: := significam literalmente tornar-se.
// Assim: (X10, 10cm) passa a ser (X10, 15cm)
// É uma forma curta de escrever
// e, em seguida, escrevemos uma palavra end (em inglês)
End → que é o fim dessa paz de código,
```

- A cor azul (início e fim) é para as palavras-chave; explicaremos isso mais tarde.
- A cor verde é para o comentário.
- O código real é simplesmente preto, com apenas uma linha.



Portanto, você pode ver que a tarefa termina aí.

Isso deixa tudo muito claro e o computador também pode ler isso.

Se quisermos aprender mais sobre programação, também devemos observar as propriedades do nosso ponto:

Ele não é apenas um DOT parado em algum lugar:

É **grosso** ou **fino** ou **pequeno** ou **grande**.

Ou é a mesma coisa??

Ele tem cor, qual delas?

Nós mesmos podemos escolher isso. Todas essas são **PROPRIEDADES**.

Um computador NÃO entende nada.

Mas ele é feito de tal forma que pode traduzir esse código e executá-lo

O **DOT** ou, em inglês, "**pixel**" é visível.

Portanto, ele tem como **propriedade: visível (visible)** ou **não visível**.

É um caso estranho: você pode dizer em uma tela que um ponto está visível ou não (ligado "**False**" ou desligado "**True**").

Mas ela está sempre presente, como em um truque de mágica.

Nomeamos isso como ligado ou desligado, como **False** ou **True**.

Significa o mesmo.

Há um homem que descobriu que essa é uma **propriedade** importante: **George Boole**.

Falaremos sobre ele mais tarde. Ele é muito importante e muito famoso.

Para um computador, isso significa que ele lê 0 ou 1.

Falso: não existe e Verdadeiro: existe

Então, um ponto que está lá, mas não podemos ver?

Isso é muito difícil de imaginar.

Vamos fingir que é apenas **transparente**.

Podemos mostrar em um exemplo de transparência:

CÓDIGO PSEUDO, (PSEUDO CODE)

Para traduzir isso em **PSEUDO CODE**, teríamos que dizer:

o ponto é visível ou não - *ligado ou desligado*.

Então, dizemos que ele está lá: com uma propriedade ligada ou desligada.

Visible := True. Ou em linguagem normal **Visível "torna-se"**:

Visible := ← (torna-se (becomes)) On → Visível (Visible) := True.

Portanto, nossa linguagem de computador **Pascal** tem uma maneira de listar todas essas **propriedades (Properties)**.

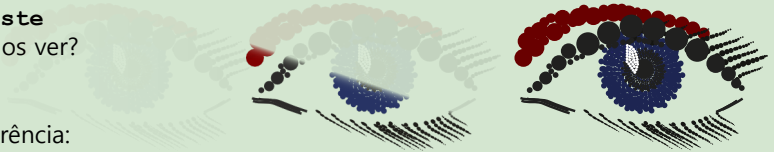
Por que explicamos tudo isso? Porque usamos com muita frequência ao criar um programa.

As PROPERTIES podem ser **reais** ou, digamos, **uma descrição - portanto, não reais**.

Na natureza, não conhecemos 15 cm. Isso **não é natural**.

Você pode pensar em uma régua para acompanhá-lo.

Mas o comprimento de algo nem sempre é exatamente 15 cm.



Você pode imaginar a extensão.

Mas não é algo que exista de fato. Você não pode pegar o comprimento.

É como você descreveu: A grama (real) é **verde** (Cor) mas também pode ser **amarelo**

Se estiver seco, por exemplo.

Portanto, a cor é uma propriedade. Você pode pegá-la.



E chamamos algo assim de **ABSTRACT**:
Ele está lá, mas às vezes você não consegue vê-lo ou compreendê-lo.
Portanto, uma **maçã** é real,
mas um **ponto de interrogação** não é.
Já descobrimos isso.



Você pode ver isso, mas você não pode levá-lo ou agarrá-lo.
O fato de algo ser visível é claro, mas é **True** ou **False**.

True or False.

Essa é uma propriedade que chamamos de ABSTRACT.

Isso significa que essa é uma maneira inventada de pensar sobre isso.

Uma ideia que significa se algo existe na realidade. Para facilitar para você: tente imaginar algo.

Para um **ponto**, isso seria um **pequeno círculo preto** ou um **quadrado** ● ■

Dizemos então: *que é um círculo com a PROPRIEDADE do preenchimento do círculo é PRETO.* (Cor)

Também pode ser de qualquer cor. Se a ponta for muito, muito pequena, você nem mesmo a verá.

Lembre-se do vídeo sobre *edição página 15 / página 6 deste artigo*

Para alguém que não enxerga bem, muitas vezes é preciso dizer que algo está lá. Eles não conseguem enxergar. É por isso que esse atributo é tão importante.

"ABSTRACT" significa que você pode imaginá-lo em sua memória e ele não precisa existir de fato.

Você pode até pensar em uma imagem com isso, o que ajuda a lembrar.

Isso facilita.

Usamos muito isso com computadores. Temos muitas dessas palavras para explicar. Mas faremos isso com a ajuda de exemplos, e eles virão.

Uma pergunta engraçada é:

O que acontece quando há dois pontos? Ou até mais?

Em uma tela de telefone ou computador, é claro, há uma quantidade enorme de pontos.

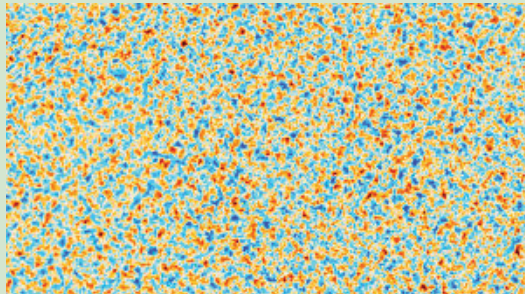
Mas todos eles estão muito próximos uns dos outros, o que torna tudo muito complicado:

They can all have their own colour.

They can be in groups together.

Think of the video example, *see issue page 15 / page 6 of this article*

For example, you can use a lot of brown dots to make a tree trunk.



9-Exemplo

Or with red dots you can make a tomato. *see issue page 15 / page 6 of this article.*

Or with very densely packed dots you can make a letter.

Try it on paper.

So you can make your own letter kind.

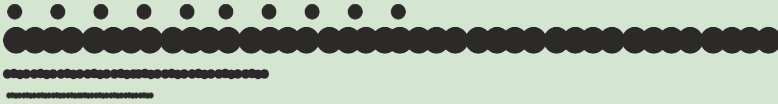
With the computer.

That's what we're going to do.



Como agora colocamos muitos pontos em uma fileira, se você olhar de longe, poderá ver uma espécie de linha se formando.

10 - Exemplo de formação de pontos de uma linha



11 - Exemplo de caracteres mesmo com ramificações.

Portanto, você também pode desenhar com ele.

Se você desenhar com um lápis na mão, essa linha geralmente oscila um pouco. É preciso praticar bastante para conseguir fazer uma linha reta ou curva bem bonita.

Com um computador, isso é muito mais fácil:

você pode instruir o computador a desenhar uma linha de um ponto a outro.

Você mesmo pode transformá-lo em um círculo, triângulo ou quadrado com um comando mais detalhado.

Então já estamos realmente programando.

Para facilitar a computação para você, criamos um pequeno aplicativo na Web no qual você pode criar seu primeiro ponto: <https://idefix.freepascal.org/~michael/pas2js-demos/turtlecompiler/>



WIKIPEDIA

O programa original chama-se **Turtle** e ainda não havia uma boa versão para Pascal.

Michael van Canneyt recriou esse modelo e o disponibilizou para o **Pascal** com a ajuda do (Pas2JS).

Falando sobre o compilador turtle:

a ideia de uma tartaruga virtual remonta à década de 1960.

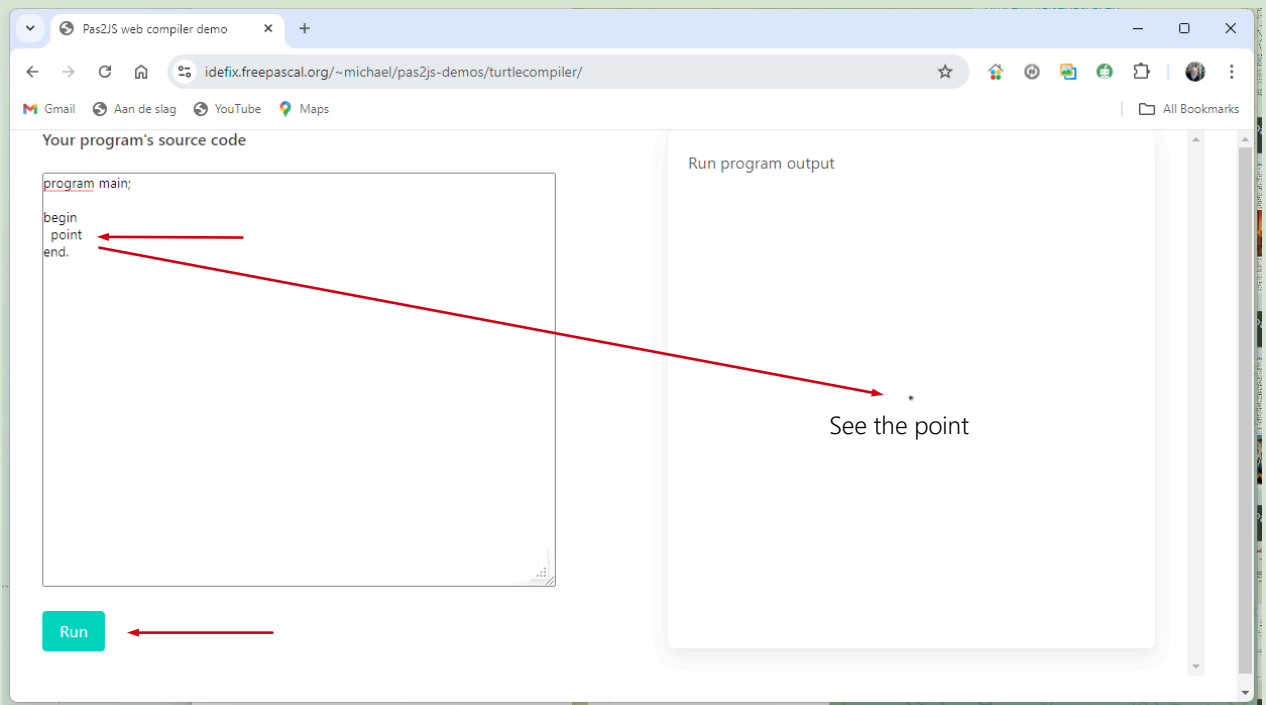
Seymour Papert e seu grupo no MIT (**Instituto de Tecnologia de Massachusetts**)


criou a linguagem de programação **LOGO** que pode controlar uma **tartaruga robô** com uma caneta física.

As crianças dirigiam o robô e ele desenhava enquanto se movia.

Seymour disse que a Tartaruga era "um objeto para pensar".

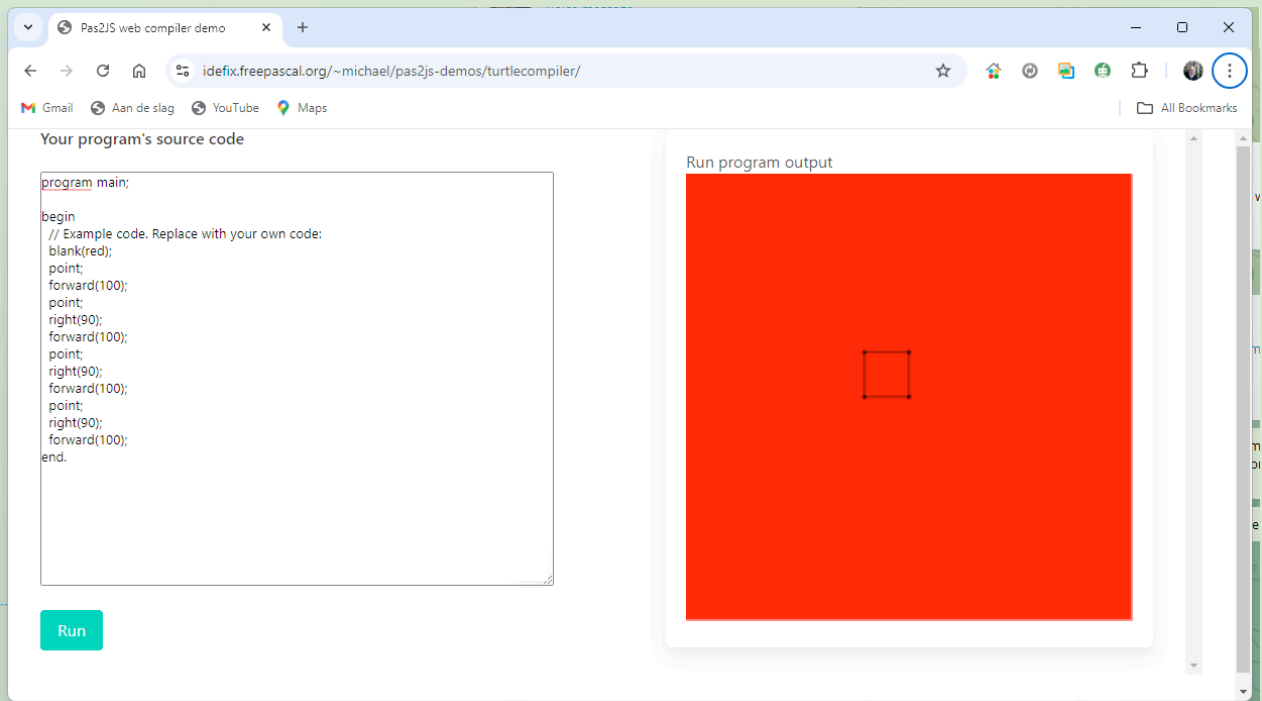
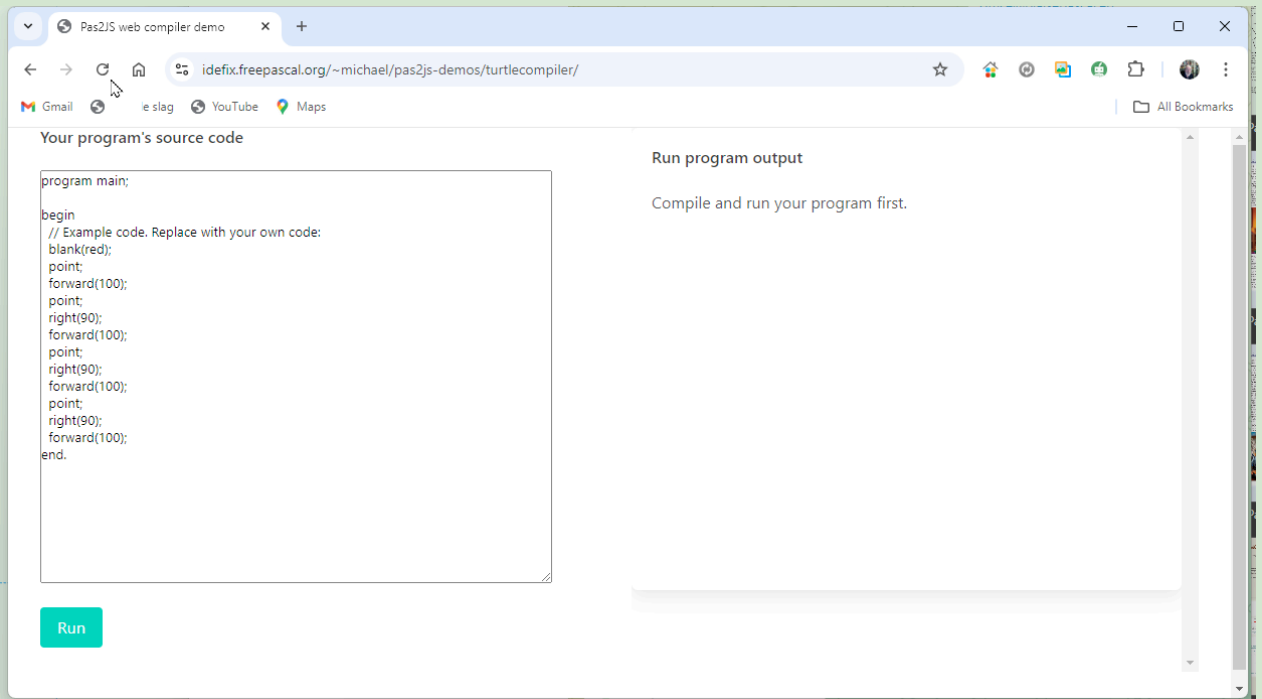




Tudo o que você precisa fazer é excluir o código fornecido por padrão e substituí-lo pelo exemplo fornecido. Na próxima página, há um exemplo que explica o que você pode fazer com a demonstração. Basta excluir um comando e executá-lo novamente. Se quiser começar do zero: clique no ícone de atualização  F5 e todas as configurações serão redefinidas desde o início



Lembre-se de que tudo isso é destinado a crianças que nunca criaram nenhum programa. Se você iniciar o compilador turtle, ele abrirá com o seguinte: simplesmente brinque com ele



Nesta página, mostraremos que outras ordens você pode dar e alguns exemplos que possibilitarão que você experimente as formas que deseja construir por conta própria.

Colors you can use
green,
red,
blue,
yellow,
violet,
lime,
orange,
skyblue,
brown,
pink,
darkgreen,
darkred,
darkblue,
ochre,
indigo,
olive,
orangered,
teal,
darkbrown,
magenta,
lightgreen,
lightred,
lightblue,
cream,
lilac,
yellowgreen,
peach,
cyan,
lightbrown,
lightpink,
seagreen,
maroon,
royal,

Colors you can use
gold,
purple,
emerald,
salmon,
turquoise,
coffee,
rose,
greengrey,
redgrey,
bluegrey,
yellowgrey,
darkgrey,
midgrey,
lightgrey,
silver,
white,
black

```
forward(n : integer);  
back(n : integer);  
left(n : integer);  
right(n : integer);  
drawxy(x,y : integer);  
movexy(x,y : integer);
```

```
e movement  
home;  
setx(x : integer);  
sety(y : integer);  
setxy(x,y : integer);  
direction(n : integer);  
angles(degrees : integer);  
turnxy(x,y : integer);
```

```
point;  
setpointsize(aSize : Integer);  
penup;  
pendown;  
colour(aColor : Integer);  
color(aColor : Integer);  
randcol(n: integer);  
gb(i : integer) : Integer;  
thickness(i : integer);
```

```
box(x,y,color : integer; border : Boolean);  
circle(radius : integer);  
blot(radius : integer);  
ellipse(xRadius,yRadius : integer);  
ellblot(xRadius,yRadius : integer);
```

```
blank(acolor : integer); example blank(white);
```

example
example
example
example
example

```
forward(100);  
back(50);  
left(90);  
right(90);  
drawxy(50,300)  
movexy(50,300)
```

The screenshot shows a web-based turtle compiler interface. On the left, a text area titled "Your program's source code" contains the following code:

```
program main;  
begin  
  // Example code. Replace with your own code:  
  blank(white);  
  setpointsize(30);  
  point;  
end.
```

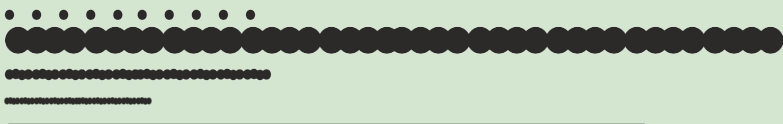
Below the code area is a green "Run" button. On the right, a window titled "Run program output" displays a single black dot. Red arrows from the code point to the dot: one from `blank(white);` to the dot, one from `setpointsize(30);` to the dot, and one from `point;` to the dot with the label "sets the point on the output".

<https://idefix.freepascal.org/~michael/pas2js-demos/turtlecompiler/>



UMA LINHA

Uma linha é uma conexão de todos os pontos entre dois pontos. Uma linha pode ser desenhada entre dois pontos de todos os tipos (quaisquer que sejam), portanto, toda linha pode ter muitos pontos (infinitos, tão grandes quanto o universo). O universo também se parece com uma linha.



O universo também se parece com uma linha.



Uma linha é plana, ou seja, é o que chamamos de unidimensional. Ela se estende infinitamente em ambas as direções e não tem espessura. Ela não é uma forma porque não tem forma. Embora possamos representar pontos ou linhas como formas porque temos que vê-los, eles não precisam ter uma forma.

Shapes



Um pixel é representado por um ponto ● ou quadrado ■ na tela de um monitor de computador ou de um telefone celular. Os pixels são uma espécie de blocos de Lego de uma imagem digital e são criados usando determinados pontos (coordenadas), como já vimos acima em nosso exemplo X10 := X15

O tamanho real de um pixel nem sempre é o mesmo, pois depende da resolução do monitor.

Resolução é uma palavra usada para telas de computador, para descrever o número de pixels usados em uma tela.

Quanto maior for esse número, maior será a **resolução máxima** da tela.

O tamanho é igual ao tamanho do espaçamento entre pontos - quando a tela está configurada com a resolução máxima.

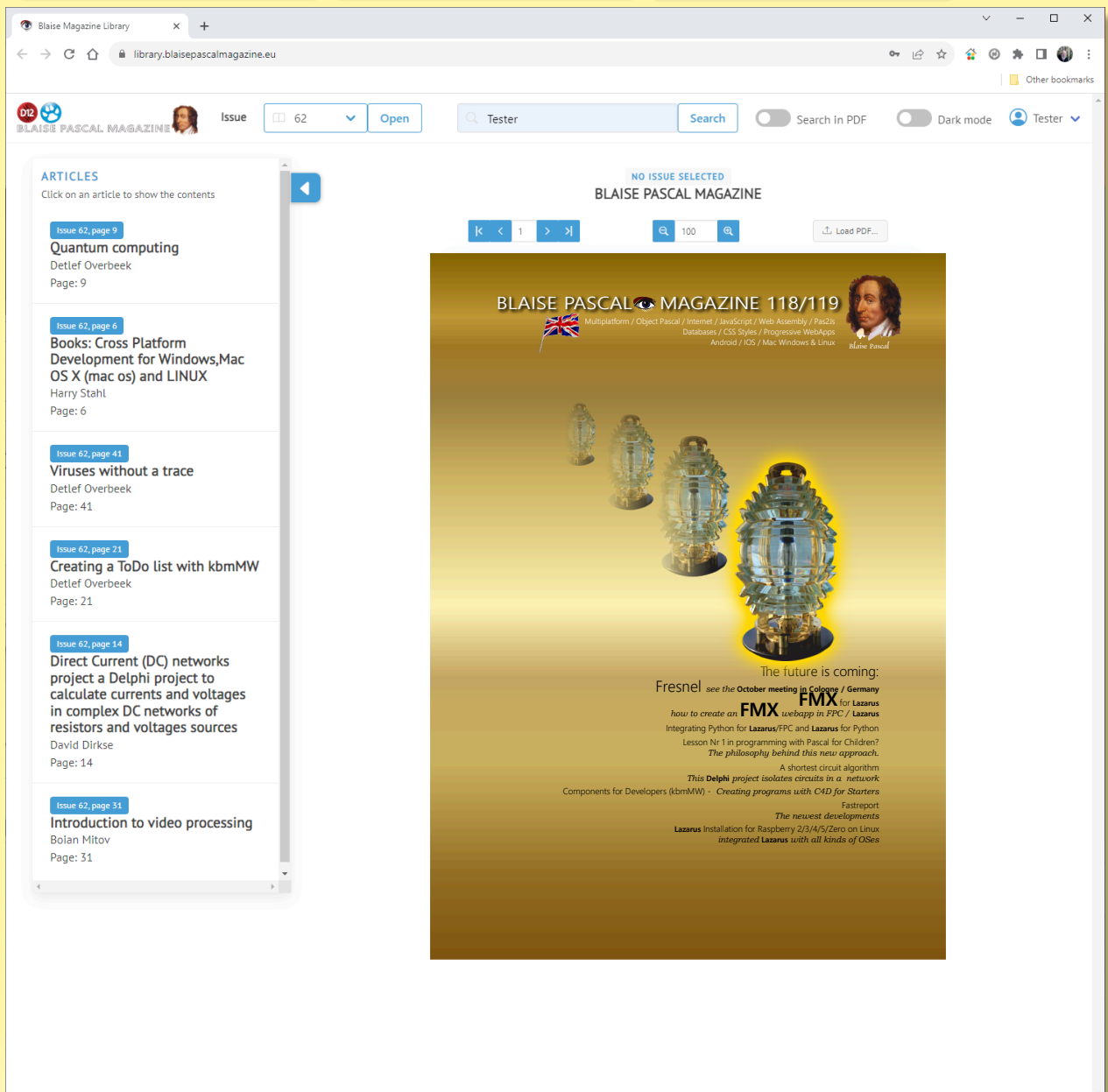
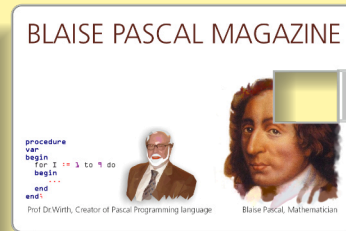
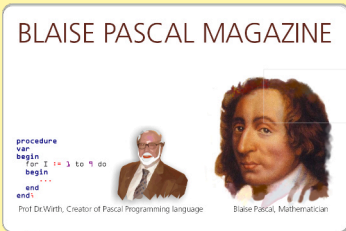
É claro que, como cada pixel usa mais pontos, não é possível criar imagens muito nítidas com poucos pixels.

Como resultado, os pixels individuais podem se tornar visíveis e, assim, você obtém desenhos em blocos e com pedaços, chamados de "**pixelados**", como pode ser visto em *página 4 deste artigo*



LIB-STICK ON USB CREDIT CARD BLAISE PASCAL MAGAZINE

LIB-STICK USB-CARD: ALL ISSUES / CODE INCLUDED. SAME INTERFACE AS THE INTERNET LIBRARY € 100

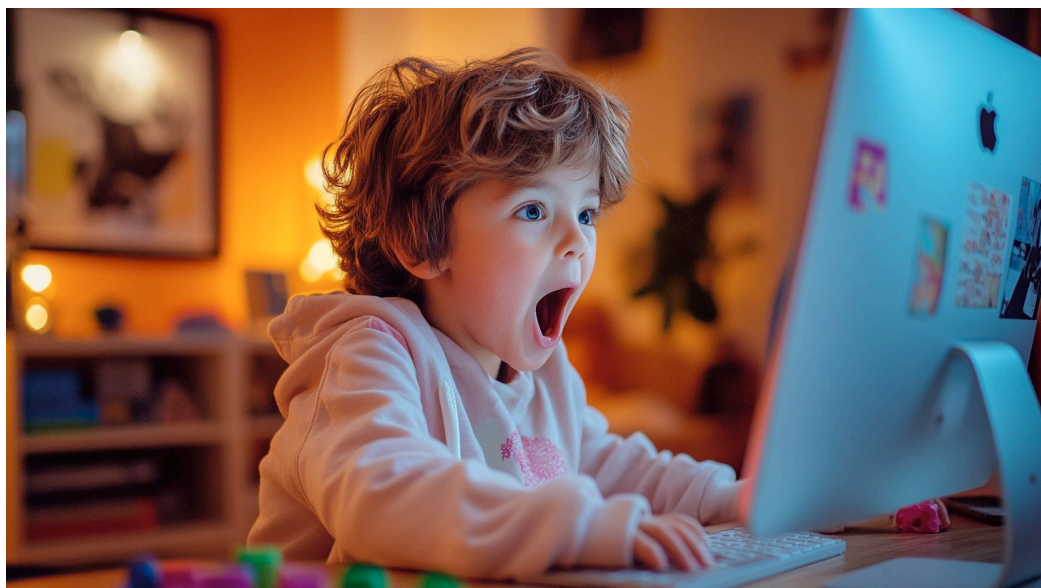


INTRODUÇÃO

O tópico deste artigo é a influência da gamificação* na maneira como as crianças aprendem! Aqui, você pode observar como os jogos e a aprendizagem interativa aumentam a motivação, estimulam as habilidades analíticas e críticas e promovem o desenvolvimento social. Isso não se aplica somente as crianças, mas também a adultos. Este artigo é uma introdução ao artigo da lição nº 1 na página 7 desta edição.



***Gamificação** é a tentativa estratégica de aprimorar sistemas, serviços, organizações e atividades criando experiências semelhantes às vividas em jogos para motivar e envolver os usuários. Isso geralmente é feito por meio da aplicação de elementos de design de jogos e princípios de jogos (dinâmica e mecânica) em contextos que não são de jogos. A gamificação faz parte do design de sistemas persuasivos e geralmente emprega elementos de design de jogos para melhorar o envolvimento do usuário, a produtividade organizacional, o fluxo, o aprendizado, o crowdsourcing, a retenção de conhecimento, o recrutamento e a avaliação de funcionários, a facilidade de uso, a utilidade dos sistemas, o exercício físico, as infrações de trânsito, a apatia dos eleitores, as atitudes do público em relação à energia alternativa e muito mais. Uma coleção de pesquisas sobre gamificação mostra que a maioria dos estudos sobre gamificação considera que ela tem efeitos positivos sobre os indivíduos. Entretanto, existem diferenças individuais e contextuais.



A MAGIA DA GAMIFICAÇÃO PARA MENTES JOVENS

É óbvio que os jogos podem ajudar seu filho a aprender melhor.

Como pais, todos nós queremos que nossos filhos gostem do processo de aprendizagem e consigam obter bons resultados em sua educação. Acontece que a incorporação de jogos interativos de aprendizagem e brincadeiras em sua rotina de aprendizagem pode desencadear um novo nível de envolvimento e diversão. Neste artigo, vamos nos aprofundar no poder do jogo e em como a gamificação pode tornar o aprendizado mais divertido e eficaz.

A IMPORTÂNCIA DO JOGO NO APRENDIZADO

Quando pensamos em brincar, geralmente imaginamos que é algo divertido e relaxante. Ela desempenha um papel fundamental no desenvolvimento e no aprendizado de seu filho ou de você mesmo.

APRENDIZADO POR MEIO DE DESCOBERTAS

Brincar permite que as crianças e nós mesmos exploremos o mundo ao nosso redor.

Seja construindo com blocos simples, fingindo ser super-heróis ou jogando jogos educativos online, as crianças aprendem usando seus sentidos, observando e experimentando.



- **DESENVOLVIMENTO COGNITIVO**

Quando as crianças brincam com quebra-cabeças, jogos de tabuleiro ou jogos de memória, elas estão exercitando suas habilidades de raciocínio, mas, o mais importante, a memória e a concentração. A concentração é uma arte que deve ser ensinada o mais cedo possível.

Suas habilidades de raciocínio são aprimoradas pelos elementos do jogo e você pode até mesmo aplicar abstrações.

É claro que isso também se aplica ao computador:

Há muitas abstrações e, se você as explicar e descrever bem, as crianças aprenderão a entender e raciocinar desde cedo.

Isso aguça sua percepção da realidade e permite que desenvolvam seus próprios pensamentos e ideias e os analisem de forma crítica.

Para um pai, é claro, isso nem sempre é fácil.

Seu filho não se deixa levar pelo caminho do jardim.

Essa capacidade de abstração acaba levando a um melhor raciocínio e, portanto, a melhores escolhas.

O desenvolvimento do cérebro é mais rápido e melhor.

Ensinar programação a crianças em um nível muito simples tem um impacto importante:

a criança aprende a pensar em abstrações e a capacidade analítica se desenvolverá rapidamente, permitindo que ela pense de forma mais consistente e clara.

- **CRESCIMENTO SOCIAL E EMOCIONAL**

Brincar com outras pessoas os ensina a cooperar, negociar e resolver conflitos.

Brincar permite que eles expressem emoções, desenvolvam empatia e pratiquem habilidades de comunicação.

- **HABILIDADES DE LINGUAGEM E COMUNICAÇÃO**

Seja fingindo, contando histórias ou brincando com fantoches, as crianças praticam o vocabulário, a comunicação e as habilidades de linguagem expressiva.

É claro que a fantasia é muito importante aqui.

Os computadores contribuem para a capacidade de leitura; se você não sabe ler, não poderá usar o computador adequadamente.

- **Brincar traz diversão e motivação para a jornada de aprendizado de seu filho.**

As crianças ficam mais engajadas, entusiasmadas e ansiosas para explorar novos conceitos quando o aprendizado parece uma brincadeira.

Eles desenvolvem uma atitude positiva em relação ao aprendizado, o que estabelece a base para uma curiosidade duradoura e o amor pela aquisição de conhecimento; é claro que o nível precisa ser orientado adequadamente.

O aprendizado sem orientação proporciona um desenvolvimento incompleto e faz com que a pessoa se sinta solitária.



ELEMENTOS DE GAMIFICAÇÃO:

CRIAR EXPERIÊNCIAS DE APRENDIZADO ENVOLVENTES

A gamificação incorpora vários elementos e estratégias para tornar o aprendizado mais envolvente e divertido. Aqui estão alguns dos principais elementos e estratégias usados na gamificação:

- **PONTOS E PRÊMIOS**

A gamificação geralmente envolve dar pontos ou recompensas às crianças quando elas concluem tarefas ou atingem marcos específicos. Os pontos podem ser acumulados para desbloquear prêmios ou passar para níveis mais altos, criando um senso de realização e progresso. Mas não deve ser exagerado:

Objetos de valor grandes certamente não devem ser colocados na linha.

As recompensas não devem causar compulsão.

- **DISTINTIVOS E CONQUISTAS**

As medalhas e conquistas do jogo são como medalhas virtuais que as crianças podem ganhar quando atingem determinados objetivos ou demonstram certas habilidades.

Essas representações visuais de sucesso dão uma sensação de realização e reconhecimento.

- **QUADROS DE LÍDERES E COMPETIÇÕES**

As tabelas de classificação criam um senso de competição amigável, permitindo que as crianças comparem seu progresso com o de outras. Esse elemento pode estimular a motivação e incentivar as crianças a se esforçarem para melhorar ao verem seus nomes subirem nas tabelas de classificação. Crie também uma tabela de classificação alternativa, não necessariamente para ser o primeiro ou o melhor, mas para saber se você resolveu o problema e o entendeu.

- **FEEDBACK E ORIENTAÇÃO**

O feedback eficaz é essencial para o aprendizado baseado em jogos. O feedback construtivo ajuda as crianças a entenderem seus pontos fortes e as áreas que precisam melhorar.

Ele as orienta no caminho certo e as motiva a continuar.

- **PERSONALIZAÇÃO**

A gamificação permite experiências de aprendizado personalizadas, adaptadas às necessidades e preferências de cada criança. As crianças podem escolher seu avatar (eu-jogo), personalizar seu caminho de aprendizado e se envolver com conteúdo que corresponda a seus interesses, tornando a experiência mais agradável e significativa.



DESBLOQUEANDO OS RESULTADOS DE APRENDIZADO DAS CRIANÇAS COM A GAMIFICAÇÃO
Vamos dar uma olhada nos efeitos positivos que ele pode ter:

① MAIOR ENVOLVIMENTO E MOTIVAÇÃO

Ao incorporar elementos de jogos, você atrai a atenção das crianças (e dos adultos) e as mantém ativamente envolvidas no aprendizado. A empolgação de ganhar pontos, desbloquear recompensas e atingir níveis mais altos alimenta a motivação delas para continuar aprendendo.

② MAIOR RETENÇÃO E COMPREENSÃO

A gamificação ajuda as crianças a se envolverem mais profundamente no conteúdo, o que também melhora a memória e a compreensão.

③ DESENVOLVIMENTO DE HABILIDADES CRÍTICAS

A gamificação promove o desenvolvimento de habilidades essenciais, como a resolução de problemas, o pensamento analítico, a tomada de decisões e a criatividade. As crianças são incentivadas a pensar estrategicamente, tomar decisões e desenvolver soluções inovadoras por meio de desafios e tarefas gamificadas.

④ AUMENTO DA COLABORAÇÃO E DAS HABILIDADES SOCIAIS

Muitas abordagens de gamificação incluem elementos colaborativos, incentivando as crianças a trabalharem com e para as outras, a se comunicarem e a compartilhar ideias. Isso promove o desenvolvimento do trabalho em equipe, da cooperação e das habilidades sociais, que são essenciais para o sucesso no mundo de hoje. Acima de tudo, trabalhar em conjunto, ou seja, colocar os interesses de todos em primeiro lugar, é muito importante para uma sociedade saudável. Fazer parte de uma equipe também é muito importante para o desenvolvimento pessoal.

⑤ Experiências positivas de aprendizado

A gamificação cria uma experiência de aprendizado positiva e agradável para as crianças. Quando o aprendizado é divertido e envolvente, as crianças desenvolvem uma atitude positiva em relação à educação. A gamificação pode ajudar muito nesse sentido.

A gamificação não se trata de substituir os métodos tradicionais de aprendizado, mas de aprimorá-los com elementos que capitalizam a tendência natural das crianças (seres humanos) de brincar e se divertir. Ao enriquecer o conteúdo educacional com elementos e estratégias semelhantes a jogos, a gamificação torna o aprendizado mais estimulante, interativo e eficaz.

DESAFIOS E CONSIDERAÇÕES SOBRE A GAMIFICAÇÃO

Embora a gamificação tenha muitos benefícios, é importante estar ciente dos possíveis desafios e considerações. Aqui estão alguns que você deve ter em mente:

■ CONCORRÊNCIA EXCESSIVA

Embora uma pequena competição amigável possa ser motivadora, um foco excessivo na competição pode levar algumas crianças ao estresse e à pressão. É essencial encontrar um equilíbrio e garantir que o ambiente de aprendizado permaneça solidário e amoroso. Portanto, a orientação desempenha um papel importante. O bullying geralmente surge da frustração.

■ POTENCIAL DE DEPENDÊNCIA

Como em qualquer atividade divertida, há o risco de tempo excessivo de tela ou de dependência excessiva do aprendizado baseado em brincadeiras. É fundamental estabelecer limites saudáveis e garantir que as crianças tenham uma abordagem equilibrada de suas atividades e responsabilidades gerais. Mas dê a elas bastante espaço.

■ CONSIDERAÇÕES ÉTICAS

A gamificação deve ser projetada levando em conta considerações éticas. É importante evitar táticas manipuladoras ou o uso de recompensas como o único motivador para o aprendizado. A ênfase deve ser sempre no incentivo ao interesse genuíno e à motivação intrínseca (uma pessoa faz algo porque obtém satisfação pessoal com isso).





Dicas práticas para implementar a gamificação em ambientes de aprendizagem infantil

Agora que entendemos os possíveis desafios, vamos dar uma olhada em algumas dicas práticas para implementar a gamificação em ambientes de aprendizagem:

- 1 **DEFINIR OBJETIVOS CLAROS DE APRENDIZADO**
Defina claramente os objetivos de aprendizagem e alinhe-os com as atividades gamificadas. Certifique-se de que os elementos e as estratégias do jogo apoiem e reforcem os resultados educacionais pretendidos.
- 2 **ESCOLHA AS FERRAMENTAS DE GAMIFICAÇÃO CERTAS**
Crie interfaces fáceis de usar, conteúdo envolvente e recursos personalizáveis.
- 3 **COLABORAR COM OS PROFESSORES**
A comunicação e a colaboração com professores e educadores são essenciais. Eles, assim como você, podem fornecer percepções, orientação e suporte para a implementação eficaz da aprendizagem gamificada na sala de aula ou em casa.

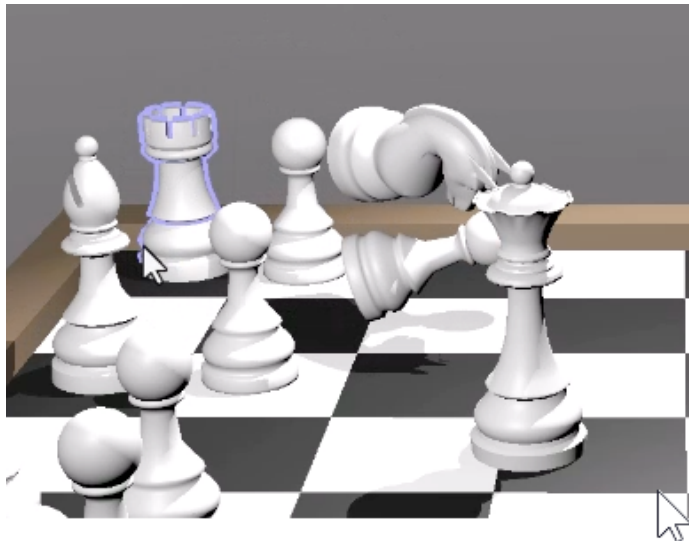
Como adultos, podemos adotar a gamificação como uma ferramenta para tornar o aprendizado mais divertido e eficaz.

Ao incorporar atividades de jogos, podemos criar um ambiente positivo e envolvente no qual nossas crianças podem prosperar e desenvolver habilidades essenciais.

COLABORAR COM OS PROFESSORES.

O aprendizado deve ser uma jornada agradável, e a gamificação pode desempenhar um papel importante até desbloquear esse prazer. Mesmo para pessoas mais velhas como eu. Na próxima edição, escreverei sobre como implementar a gamificação em sua interface de usuário.

Com a ajuda de Michalis Kamburelis, o fundador do Castle Game Engine



We are GDK.

Do you have a Delphi challenge and are you looking for expertise or capacity? Our experts and developers are ready to realise your ambitions and goals.



30+

Delphi developers

Work with our Delphi Experts

99+

Delphi conversions

Smart upgrade to the latest Delphi

5

Embarcadero MVPs

Authority within the Delphi community

4

Offices worldwide

The Netherlands, UK, Brazil and USA



GDK IN A NUTSHELL

About GDK.

We share a passion for software development and love to keep up with the latest technologies. We have a strong team of specialists with a lot of knowledge and expertise in Delphi.

ACHIEVING YOUR AMBITIONS

Work together .

Looking for a partner to maintain and extend your software? Or upgrade your software to the newest Delphi version? We are ready to help you!

We are Delphi

Contact us

www.gdksoftware.com



POCKET PACKAGE (2BOOKS)

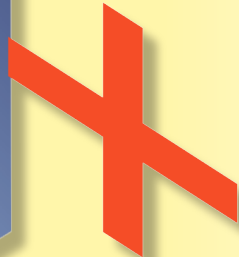
PRICE: € 25,00

EXCLUDING VAT AND SHIPPING

Lazarus HANDBOOK POCKET



**SPECIAL SALES
PRICE € 25
EX VAT
EX SHIPPING**



<https://www.blaisepascalmagazine.eu/product-category/books/>





Starter Expert

By Reinhold Schmitt



INTRODUÇÃO

Também adorei a motivação da **Raspberry Pi Foundation** de criar um computador pequeno e barato para treinar jovens ou para a cena dos criadores com um sistema operacional **Linux** completo (*Anteriormente raspbian, agora Raspi OS*).

Então, comprei o **Raspberry Pi 1 Versão 2 Modelo B com 512 MB de memória RAM**.

Agora, é claro, eu queria programá-lo.

Na época, Python era estranho para mim e eu não queria ter nada a ver com **C** e **C++**.

Portanto, tive que instalar o compilador **FreePascal**. Agora eu podia escrever meus primeiros programas **Pascal** com o editor **Geany** e o compilador, e também experimentar os **GPIOs**.

Como o Raspi é predestinado para aquisição de dados devido ao seu tamanho e baixo consumo de energia, recebi uma bomba de calor em 2013. Meu primeiro projeto (*maior*) com o **Raspi** foi medir as temperaturas (*com o sensor DS1820*) do gás de aquecimento a cada minuto na entrada e na saída do tanque de armazenamento de água.

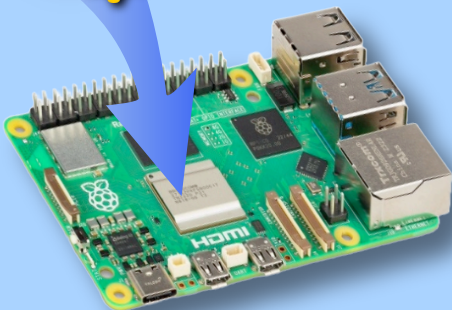
Obviamente, os dados também devem ser exibidos de forma agradável.

Para escrever um programa fácil de usar, foi necessário criar uma interface gráfica agradável.

O que poderia ser mais adequado do que **Lazarus**?

Não me lembro se instalei o **Lazarus** a partir dos repositórios ou se o construí a partir dos códigos-fonte para o **Raspi** (*make bigide*).

De qualquer forma, o arquivo de troca do **Raspi** teve que ser aumentado e o trabalho com o **Lazarus** ficou muito, muito lento. Você também podia "ir tomar um café" enquanto compilava.



Então, escrevi meu programa **Lazarus** no **PC** com **Linux** o máximo que pude e depois transferei o projeto para o **Raspi** e fiz o "ajuste fino" e o compilei lá.

Em 2015, o **Raspberry Pi 2** foi lançado com 4 núcleos e 1 GB de RAM.

Isso foi acompanhado por uma melhoria notável ao trabalhar com o **Lazarus**. Com o **Raspberry Pi 3** (2016) com **Wi-Fi** integrado (*antes era necessário um dispositivo Wi-Fi*), foi obtido outro ganho significativo de desempenho.

Com as pequenas deficiências (*Por exemplo, os tremas e os colchetes {äöüÄÖÜß} não pôde ser inserido diretamente pelo teclado; solução alternativa: via AltGr XXX ou cópia em*)- Você só precisava se acostumar com isso.

Até então, eu usava **2 Raspi 2 e 3** cada um no meu empregador para tarefas especiais de aquisição e controle de dados de medição com "tarefas ininterruptas" com o **Lazarus**.

Maravilhoso.

O **Raspberry Pi 4** está disponível desde 2019 e o **Pi 5** com **4 e 8 GB de RAM** desde 2023. Com esses **Raspi**, agora você pode trabalhar sem problemas com o **Lazarus**. Não há (*quase*) nenhum desejo não realizado e foi muito divertido. No entanto, a instalação não foi tão fácil.





INSTALAÇÃO DO Lazarus NO RASPBERRY PI

A instalação do **Lazarus** foi bastante problemática.

As versões do **Lazarus** dos repositórios raspbian eram, em sua maioria, antigas, de modo que meus projetos do **Lazarus** do PC com a versão estável atual no Raspi (**muitas vezes**) não puderam ser compilados.

As descrições de instalação em vários fóruns e **wikis** referiam-se, em sua maioria, a versões do tronco e, muitas vezes, estavam desatualizadas (*A data de criação geralmente também não estava disponível*). Em geral, o **FPC** era fácil de instalar no **Raspi**. Mas, para instalar o **Lazarus**, era preciso encontrar as bibliotecas que faltavam (*libs*) (*a partir das descrições nos fóruns ou wikis*).

A instalação do **Lazarus** atual (*estável*) para **Debian**, **Linux Mint**, **Ubuntu** etc. (pacotes deb) ou **OpenSuse**, **Fedora** etc. (pacotes rpm) e para Windows (*para CPUs Intel/AMD*) sempre foi muito fácil, em contraste com a instalação no **Raspberry Pi** com processadores **ARM**.

Faça o download de três pacotes e instale-os na ordem correta (*sempre fpc, fpc-src e Lazarus*) e a instalação estará concluída.

O programa '**fpcupdeluxe**' (*de DonAlfredo*) também está disponível há alguns anos.

Essa ferramenta permite não apenas a instalação do **Lazarus** em todas as plataformas possíveis, incluindo o **Raspberry Pi**, mas também a instalação de crosscompilers.

Por mais abrangente que seja esse programa, considero-o muito complexo e sobrecarregado para iniciantes.

Mas recentemente foram atualizados os pacotes deb para a instalação do **Lazarus** no Raspberry Pi. (criado por Otto D).

- **fpc-laz_3.2.2-240529_arm64.deb**
- **fpc-src_3.2.2-240529_arm64.deb**
- **Lazarus-project_3.4.0-0_arm64.deb**

Com esses arquivos, consegui executar o **Lazarus** no Raspberry Pis 3,4,5 e no Zero2 (com 2024-07-04-raspbios-bookworm-arm64 e 2024-03-12-raspbios-bookworm-arm64 respectivamente).

INSTRUÇÕES PARA A INSTALAÇÃO DO Lazarus NO RASPBERRY PI 3,4,5 OU ZERO 2

OS no Raspberry Pi: Bookworm 64 bit

Observação:

Por padrão, a subestrutura gráfica do **Bookworm** é definida como **Wayland**.

Para instalar o **Lazarus**, ele deve estar configurado para x11.

Verifique qual gerenciador de exibição:

```
no terminal: echo $XDG_SESSION_TYPES --> Wayland or x11
```

(o **Raspi** deve ser conectado diretamente ao monitor, sem operação remota)

Mudar do **Wayland** para o x11:

Inicie a ferramenta de configuração do software **Raspberry Pi** com "sudo raspi-config" no terminal.

Selecione '6 Advanced Options'

Selecione 'A6 Wayland' e depois W1 x11 → Reboot

Faça o download dos três arquivos **FPC/Lazarus-deb**:

LINK PARA DOWNLOAD:

<https://sourceforge.net/projects/Lazarus/files/Lazarus%20Linux%20arm64%20DEB/Lazarus%204.0RC1/>

Instalação do **fpc/Lazarus-deb** pacotes no terminal

```
sudo apt install /path-of-deb-deb-Pakete/fpc-laz-xxxx_arm64.deb
```

```
sudo apt install /path-of-deb-deb-Pakete/fpc-src-xxxx_arm64.deb
```

```
sudo apt install /path-of-deb-packages/Lazarus-project-xxxx_arm64.deb
```

NOTA SOBRE O DEBBUGER:

No momento, o depurador deve ser definido como o depurador **GNU (Gdb)** para cada novo projeto. Para fazer isso, vá para

Project → **Project settings** → **Debugger** → **Debugger backend** → **Select Gdb**.





CONCLUSÃO

Por mais de 9 anos, meu primeiro **Raspberry Pi** (*Pi 1*) tem feito seu trabalho no porão e medido temperaturas (24/7). Apenas um sensor de temperatura desistiu após 9 anos. Afinal de contas, ele foi "envelhecido" entre temperaturas de 20 a 80 °C. O stick de **WLAN** também não sobreviveu ao verão quente de 2023. Durante 6 meses, o antigo **Raspi** funcionou sem uma interface gráfica de usuário e eu escrevi um programa **Python3** para registrar a temperatura. Transformei o antigo programa **Lazarus** em um programa para exibir minhas temperaturas de aquecimento e dados de medição **fotovoltaica**.

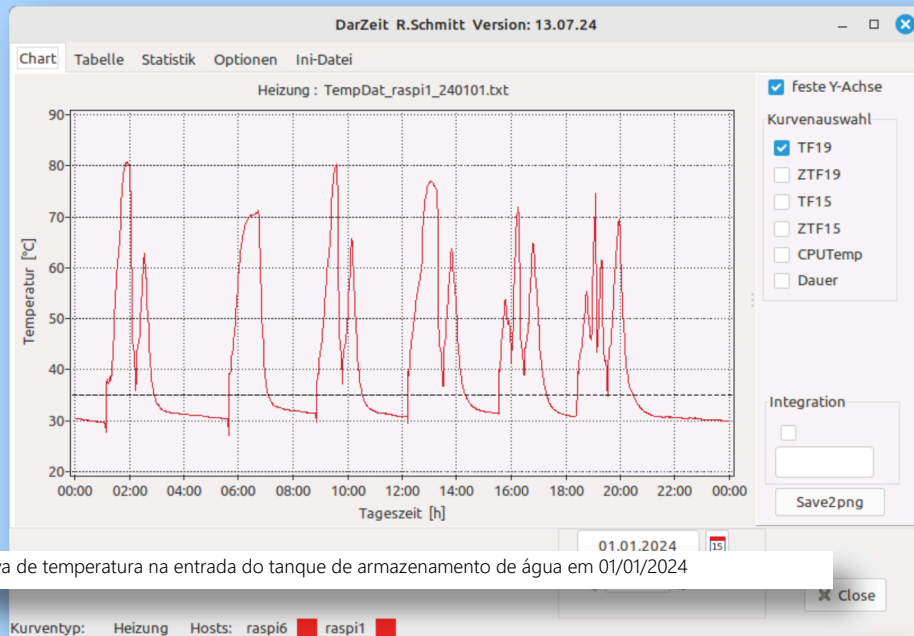


Figura 1 Curva de temperatura na entrada do tanque de armazenamento de água em 01/01/2024

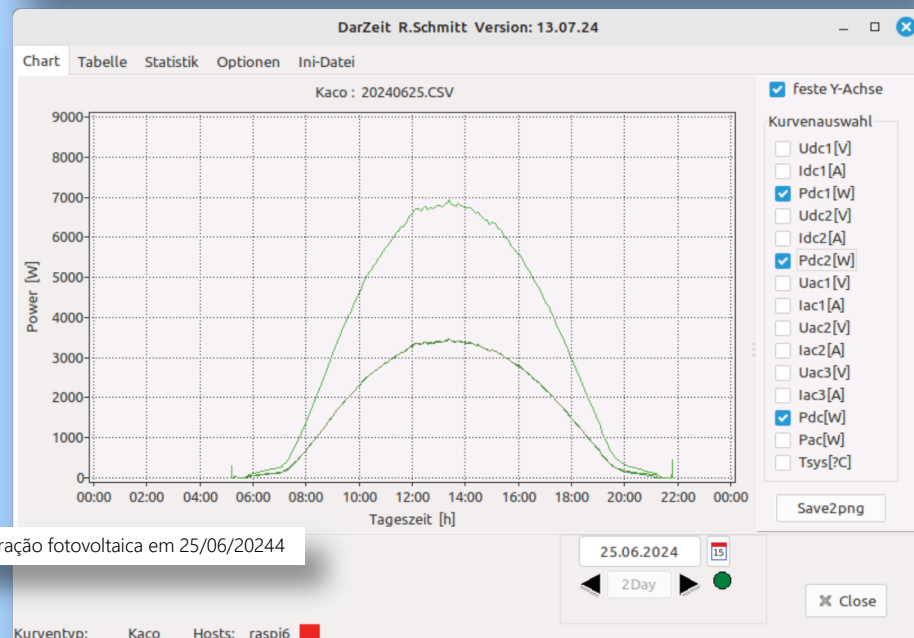
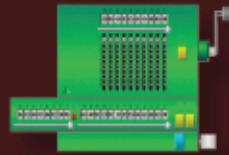


Figura 2 Geração fotovoltaica em 25/06/20244



ADVERTISEMENT

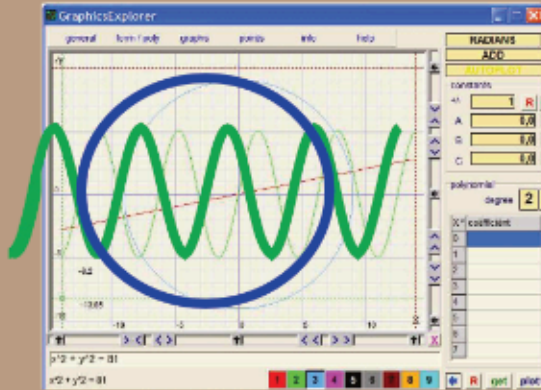
David Dirkse's website: davdata.nl/math



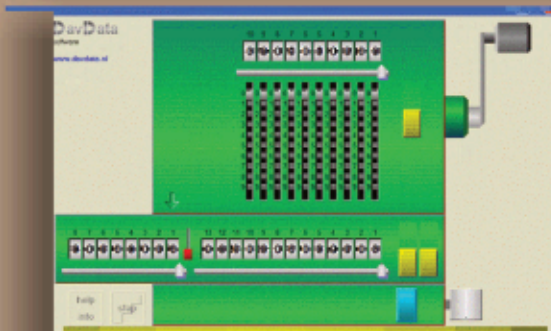
DAVID DIRKSE

including 50 example projects

<https://www.blaisepascalmagazine.eu/en/product/books-computer-graphics-math-games-download-pdf/>



**PDF EDITION
SPECIAL
OFFER € 35**



COMPUTER (GRAPHICS) MATH & GAMES IN PASCAL

<https://www.blaisepascalmagazine.eu/product-category/books/>





DO SEU EDITOR:

Acho que o PFAS é uma das maiores ameaças à espécie humana, pior até do que a guerra. Espero que você, como leitor, se interesse por esse assunto como eu.

INTRODUÇÃO

A abreviação PFAS significa substâncias per e polifluoroalquil.

Essas substâncias são produzidas pelo homem e não ocorrem naturalmente no meio ambiente.

Exemplos de PFAS são o GenX, o ácido perfluorooctanóico PFOA e os perfluorooctanossulfonatos PFOS.

Os PFAS são usados em muitos produtos.

Como resultado, e devido a emissões e incidentes, essas substâncias foram parar no meio ambiente e agora são encontradas, entre outras coisas, no solo, em resíduos de dragagem e em águas superficiais e, portanto, em nosso corpo.



WIKIPÉDIA

O QUE É PFAS?

PFAS é um nome coletivo que significa substâncias per e polifluoroalquílicas.

Esse grupo de substâncias químicas é produzido pelo homem e não ocorre naturalmente no meio ambiente. Os PFAS podem ter um efeito negativo sobre o meio ambiente e a saúde.

Exemplos bem conhecidos de PFAS são o ácido perfluoro octanóico PFOA, os sulfonatos de perfluorooctano PFOS (ácido perfluorooctano sulfônico) e as substâncias GenX.

O número exato de diferentes PFAS produzidos por seres humanos não é conhecido.

A Organização para Cooperação e Desenvolvimento Econômico (OCDE) identificou mais de 4.000 PFAS, mas pode haver mais.

Também não há clareza na ciência sobre quais substâncias são exatamente PFAS e quais não são.

PARA QUE OS PFAS SÃO USADOS?

Os PFAS têm propriedades úteis: são repelentes de água, graxa e sujeira.

Eles são encontrados em vários produtos, como notas adesivas (a cola), incluindo lubrificantes, materiais de embalagem de alimentos, espuma extintora, revestimentos antiaderentes em painéis, roupas, tecidos e cosméticos. Eles também são usados em várias aplicações e processos industriais. Os PFAS também podem ser usados de outras formas.

As substâncias são liberadas no meio ambiente por meio de emissões de fábricas que produzem ou usam as substâncias.

Também podem ser liberadas no meio ambiente por meio do uso de produtos que contêm PFAS, como espumas de combate a incêndio, agentes de impregnação têxtil, lubrificantes ou produtos que contêm PFAS, que podem ser descartados como resíduos.



O PROCESSO DE DESENVOLVIMENTO DE UM REMÉDIO CONSISTE EM DUAS ETAPAS.

Em outras palavras, a primeira etapa é a coleta do material, onde ele pode ser extraído de qualquer substância. Para fins de divisão, é útil saber que quantidades substanciais dele estão presentes na água e, conseqüentemente, no solo.

A segunda etapa é: após a extração do PFAS de um material, a etapa seguinte será a quebra das moléculas.

Neste artigo, tentarei explicar alguns métodos que conheci.

O problema de destruí-lo é que ele foi projetado para ser indestrutível.

O problema da poluição da água causada pelos produtos químicos usados na tecnologia atual está se espalhando rapidamente pelo mundo.

De acordo com um estudo recente do Centro de Controle de Doenças dos EUA, 98% das pessoas testadas tinham níveis detectáveis de PFAS na corrente sanguínea.

Os PFAS são uma família de compostos que têm vida particularmente longa e também são conhecidos como "produtos químicos eternos". Como diziam os gregos antigos: A caixa de Pandora está aberta.

Como podemos resolver esse absurdo?

Um novo material de filtragem, desenvolvido por pesquisadores do MIT, tem o potencial de oferecer um remédio natural para esse problema persistente de contaminação.

O material, que é composto de seda e celulose naturais, tem a capacidade de eliminar uma grande variedade desses compostos persistentes, além de metais pesados.

Além disso, suas propriedades antimicrobianas podem ajudar a evitar que os filtros fiquem sujos.



Fibra de seda natural

Imagem obtida de https://www.researchgate.net/figure/Natural-Silk-Fiber-34_fig6_350431006



Cellulose: Image sourced from <https://breinërco.com/3-properties-of-cellulose-gaskets-2/>

Cosméticos, embalagens de alimentos, roupas resistentes à água, espumas de combate a incêndios e revestimentos antiaderentes para painéis estão entre os inúmeros produtos que contêm compostos de PFAS.

Somente nos Estados Unidos, 57.000 locais foram identificados como contaminados por esses compostos em um estudo recente. Para cumprir as novas regulamentações que determinam um limite de menos de 7 partes por trilhão de PFAS na água potável, a Agência de Proteção Ambiental dos EUA estimou que a remediação de PFAS aumentará para uma despesa anual de US\$ 1,5 bilhão. (Se você for bem-sucedido).

A contaminação por PFAS e compostos similares é de fato um problema muito significativo, e as soluções atuais podem resolver apenas parcialmente esse problema de forma muito eficiente ou econômica. A solução desenvolvida pelo MIT se baseia em celulose e proteínas.

O MIT desenvolveu esta solução que é totalmente natural e é baseada em celulose e proteína.

Um dos funcionários da equipe, o Sr. Marelli, explicou: "Fomos apresentados ao projeto por acaso". Inicialmente, seu grupo desenvolveu a tecnologia que possibilitou o material de filtragem para uma finalidade totalmente não relacionada: criar um sistema de rotulagem para combater a proliferação de sementes falsificadas, que frequentemente são de qualidade inferior.

Sua equipe desenvolveu um método para transformar as proteínas da seda em cristais uniformes em nanoescala,

ou "nanofibrilas", usando uma técnica de fundição de gotas à base de água em temperatura ambiente que não agride o meio ambiente.

Eles postularam que seu novo material nanofibrilar poderia ser eficaz na filtragem de contaminantes; no entanto, seus esforços iniciais apenas com as nanofibrilas de seda não foram bem-sucedidos.

A equipe optou por incorporar uma substância adicional: a celulose, que é facilmente acessível e pode ser extraída de resíduos de fibras de madeira agrícola.

Os pesquisadores empregaram um método de automontagem no qual a proteína da fibroína da seda é suspensa em água e, posteriormente, modelada em nanofibrilas por meio da inserção de "sementes" de nanocristais de celulose. Isso faz com que as moléculas de seda, anteriormente desorganizadas, se alinhem ao longo das sementes, estabelecendo, assim, a base de um material híbrido com propriedades exclusivas.





Imagem obtida de <https://images.squarespace-cdn.com/content/v1/629780ad60b2ca4521d4fcbe/9180091f-bd92-4ead-aeac-db47ac902fa2/handwashing.jpg?format=2500w>

Os pesquisadores desenvolveram um material altamente eficaz na remoção de contaminantes em experimentos de laboratório, integrando a celulose em fibrilas à base de seda que poderiam ser formadas em uma membrana fina e, posteriormente, ajustando a carga elétrica da celulose. As nanofibrilas eram ineficazes isoladamente.

A equipe optou por incorporar uma substância adicional:

a celulose, que é facilmente acessível e pode ser extraída de resíduos de fibra de madeira agrícola.

Os pesquisadores empregaram um método de automontagem no qual a proteína da fibroína da seda é suspensa em água e, posteriormente, modelada em nanofibrilas por meio da inserção de "sementes" de nanocristais de celulose. Isso faz com que as moléculas de seda, anteriormente desorganizadas, se alinhem ao longo das sementes, estabelecendo assim a base de um material híbrido com propriedades exclusivas.

Eles também descobriram que a celulose possuía propriedades antimicrobianas potentes devido à sua carga elétrica.

Essa é uma vantagem significativa, pois a contaminação por bactérias e fungos é uma das principais causas de falha da membrana de filtragem. Os pesquisadores afirmam que o problema da contaminação deve ser significativamente atenuado pelas propriedades antimicrobianas desse material.

De acordo com Marelli, esses materiais são capazes de competir com os materiais padrão atuais na filtragem de água em termos de extração de íons metálicos e desses contaminantes emergentes, e também podem superar alguns deles em uma base atual.

Os materiais foram capazes de extrair ordens de magnitude a mais de contaminantes da água em experimentos de laboratório do que os materiais padrão atuais, carvão ativado ou carvão ativado granular.

De acordo com Marelli, a equipe pretende continuar aprimorando o material, especialmente em termos de durabilidade e disponibilidade de materiais de origem, apesar do fato de que o novo trabalho funciona como uma prova de princípio.

Embora as proteínas da seda utilizadas neste estudo sejam acessíveis como subproduto do setor têxtil da seda, o suprimento pode ser inadequado se esse material for expandido para atender à demanda global por tecnologias de filtragem de água. Além disso, é possível que materiais proteicos alternativos sejam capazes de cumprir a mesma função a um custo menor.

O material será empregado inicialmente como um filtro de ponto de uso, que poderia ser fixado em uma torneira de cozinha. Eventualmente, ele poderá ser expandido para fornecer filtragem para o abastecimento de água municipal;

No entanto, isso deve ser feito somente depois que os testes confirmarem que não há perigo de introduzir qualquer contaminação no abastecimento de água.

Eles afirmam que a maior vantagem do material é que os constituintes da seda e da celulose são classificados como substâncias de grau alimentício, o que torna a contaminação altamente improvável.

O texto original pode ser encontrado aqui (*basta clicar*)

UMA POSSÍVEL SOLUÇÃO REALISTA.

Usando luz LED visível, pesquisadores da **Ritsumeikan, em Kyoto, Japão**, afirmam ter criado uma maneira revolucionária e **ecologicamente** correta de erradicar substâncias perigosas. Um poluente persistente conhecido como **PFAS** foi praticamente totalmente decomposto à temperatura ambiente por essa técnica. Esse novo método resolve problemas ambientais e econômicos ao mesmo tempo, recuperando o precioso flúor e removendo simultaneamente esses contaminantes de longa duração.

(O flúor tem muitos usos. Ele é encontrado como flúor na pasta de dente e na água potável, no Teflon, em medicamentos, incluindo o medicamento quimioterápico fluorouracil, e no ácido fluorídrico. Ele é usado em refrigerantes (clorofluorcarbonos ou CFCs), propelentes e para o enriquecimento de urânio pelo gás UF6. O flúor não é um elemento essencial na nutrição humana ou animal. Acreditava-se que a aplicação tópica de flúor, como na pasta de dente ou no enxaguante bucal, era eficaz para a conversão da hidroxiapatita do esmalte dentário em fluorapatita mais forte, mas estudos mais recentes indicam que o flúor ajuda no crescimento do esmalte).

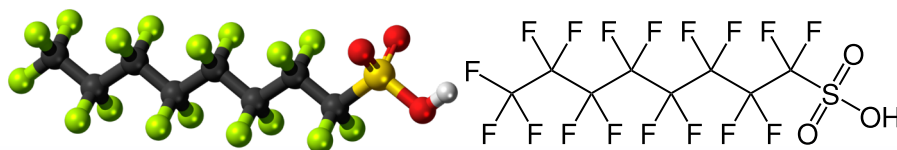
Com esse método inovador, um PFAS pode ser completamente decomposto em oito horas, enquanto o fluoropolímero Nafion pode ser decomposto em 81% em vinte e quatro horas.

O principal autor do estudo, o **professor Yoichi Kobayashi**, expressou otimismo quanto ao potencial da metodologia para decompor várias substâncias **perfluoroalquílicas** em um ambiente seguro. Ele acredita que isso ajudará a pavimentar o caminho para uma sociedade que possa reciclar o **flúor**. A abordagem atual de decomposição de compostos eternos utiliza calor - uma grande quantidade de calor, na verdade - caso você não tenha percebido.

A temperatura é de 752 graus **Fahrenheit** (400 graus **Celsius**), de acordo com o padrão. Atualmente, a queima de combustíveis fósseis, que têm seus próprios problemas ambientais e de saúde, é normalmente o único método para atingir esse tipo de temperatura.

Pesquisadores japoneses desenvolveram uma maneira de degradar produtos químicos que duram para sempre, iluminando com luz LED visível os nanocristais em uma solução que também contém PFAS, FPs e TEOA.

Os cientistas descobriram que as fortes ligações de carbono-flúor das moléculas de PFAS podem ser **quebradas pela irradiação de nanocristais semicondutores, que produzem elétrons com um alto potencial de redução**.



CONCLUSÃO:

A cadeia inquebrável da molécula está sendo alterada por dois fatores:

- ① adição de calor (que torna a cadeia de moléculas mais fácil de ser atacada) e
- ② do que a força em um elétron, o que significa que a cadeia é quebrada.

o texto original pode ser encontrado aqui (*basta clicar*)



Lazarus HANDBOOK SPECIAL SALES

PRICE: € 25,00

POCKET PACK (2 BOOKS)

EXCLUDING SHIPPING





Delphi LSP: 64-BIT POWER Delphi LSP: POTÊNCIA DE 64 BITS PARA GRANDES PROJETOS E CODEINSIGHT APRIMORADO

A versão mais recente do RAD Studio apresenta algumas melhorias notáveis no **Delphi** Language Server Protocol (LSP), facilitando a vida dos desenvolvedores que trabalham com grandes bases de código. Uma das maiores atualizações é a versão de 64 bits do mecanismo **DelphiLSP**, que elimina os limites de memória e proporciona um desempenho mais suave para grandes projetos.

64-BIT LSP: NÃO HÁ MAIS LIMITES DE MEMÓRIA

A versão de 64 bits do mecanismo **DelphiLSP** foi projetada para lidar com projetos enormes sem as restrições de memória do mecanismo de 32 bits. Se você já teve problemas de lentidão no desempenho ou travamentos com aplicativos grandes, essa nova versão é um divisor de águas. É fácil de ativar no IDE e garante uma experiência muito mais suave com o CodeInsight.

MELHORIAS DE QUALIDADE PARA CODEINSIGHT

Juntamente com a atualização de 64 bits, o mecanismo **DelphiLSP** traz vários aprimoramentos destinados a tornar a codificação mais intuitiva:

- **Symbol Hints:** As dicas agora funcionam para a maioria dos símbolos nas interfaces de classe, incluindo campos, proporcionando acesso rápido às informações sobre os membros da classe.
- **Local Constants and Variables:** Passar o mouse sobre constantes e variáveis locais agora mostra suas definições, facilitando o rastreamento de valores sem sair do código.
- **Forwarded Type Declarations:** O mecanismo agora pode resolver declarações de tipo encaminhadas em mais casos, reduzindo erros e melhorando a navegação no código.

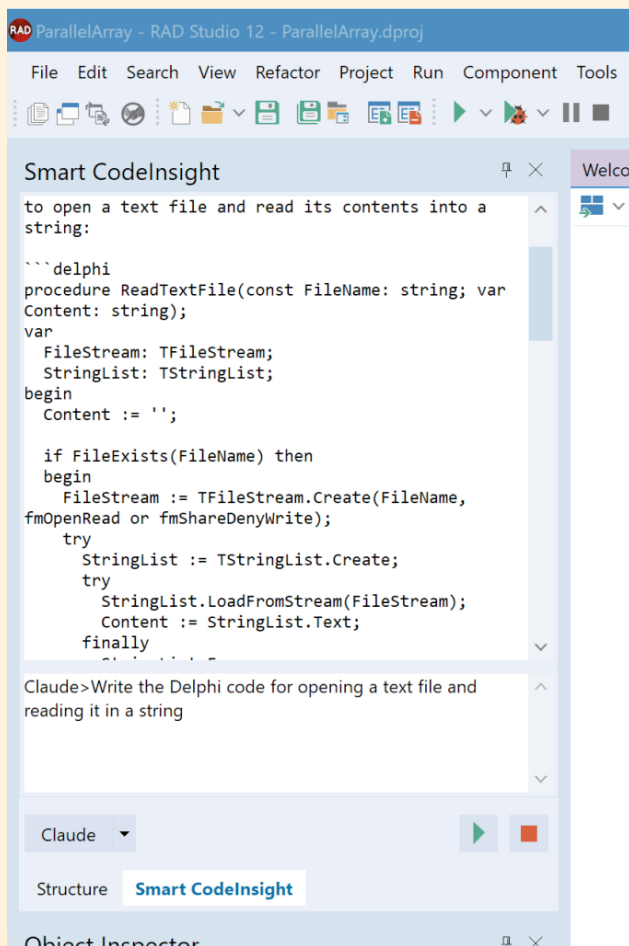
CONCLUSÃO

O novo mecanismo **DelphiLSP** de 64 bits e seus recursos aprimorados do CodeInsight destinam-se a aplicativos muito grandes, e é por isso que ele está presente nas edições Enterprise ou Architect do RAD Studio. Com desempenho aprimorado para grandes bases de código e manipulação de símbolos mais inteligente, essas atualizações ajudam a simplificar seu fluxo de trabalho e a manter um código limpo e eficiente.





❶ USE O ACESSO INCORPORADO A LLMs PARA UMA CODIFICAÇÃO MELHOR E MAIS RÁPIDA COM ASSISTÊNCIA DE IA (SMART CODEINSIGHT)
RAD Studio 12.2 apresenta uma arquitetura aberta para plug-ins de IA, oferecendo suporte pronto para uso a três soluções on-line (OpenAI, Gemini do Google e Claude da Anthropic) e uma solução off-line (Ollama). O suporte à IA do RAD Studio é chamado de Smart CodeInsight e se baseia em uma arquitetura aberta, alimentada por novas interfaces na API de ferramentas do IDE. O Smart CodeInsight também fornece recursos prontos para uso, como uma janela de bate-papo de IA e comandos de menu do editor para invocar operações de IA.





What's New in <RAD Studio 12.2 Athens>

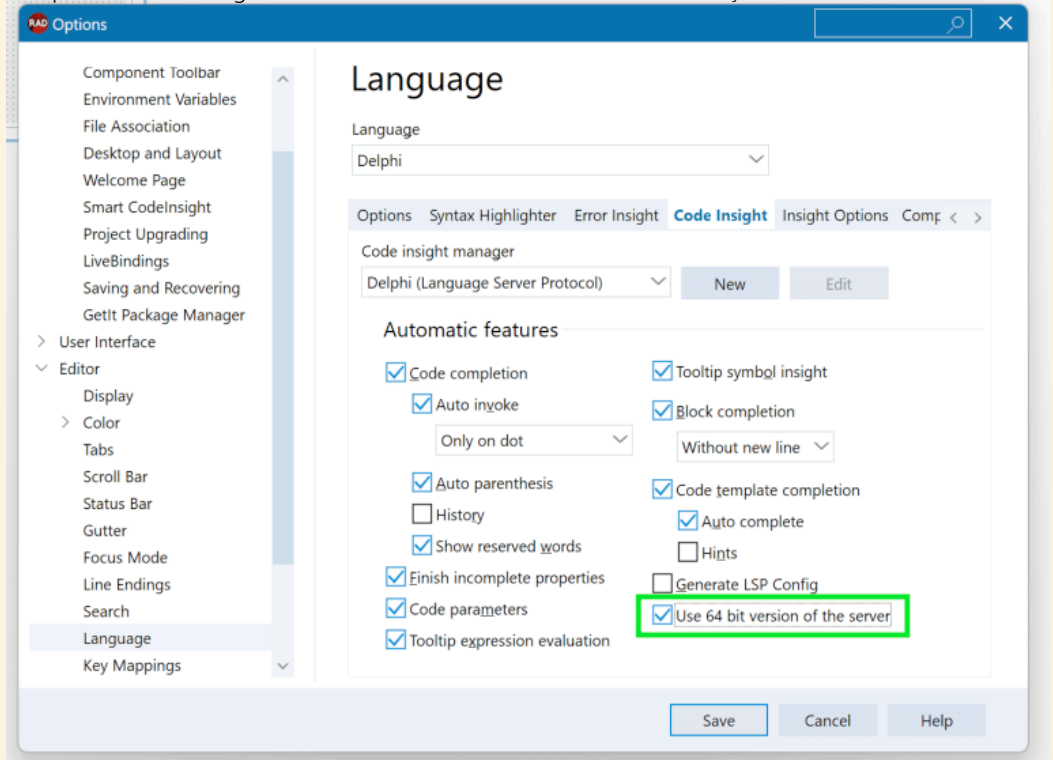
The new 64-bit compiler and LSP mark an important milestone for Delphi. Not only do they remove memory constraints when working with large projects, but they also bring us one step closer to a full 64-bit IDE.

Dalija Prasnikar
Book Author, Consultant, Embarcadero MVP

SEE WHAT'S NEW

❷ **CRIE APLICATIVOS MUITO MAIORES E MUITO MAIS RÁPIDOS COM A VERSÃO DE 64 BITS DO MECANISMO DelphiLSP**

Se você trabalha com grandes bases de código, tudo ficará muito mais fácil com a versão 12.2. A nova versão apresenta uma versão de 64 bits do mecanismo **DelphiLSP** que elimina os limites de memória e proporciona um desempenho mais suave para grandes projetos, permitindo que os usuários criem aplicativos extremamente grandes como um único executável. A versão de 64 bits do mecanismo **DelphiLSP** foi projetada para lidar com projetos enormes sem as restrições de memória do mecanismo de 32 bits. Isso é um divisor de águas para os desenvolvedores que conhecem as frustrações associadas a lentidões de desempenho ou travamentos com aplicativos grandes. Os novos compiladores estão atualmente disponíveis na linha de comando e no IDE ao usar a compilação externa. Eles só estão incluídos como parte das edições Enterprise e Architects, devido ao foco em clientes com bases de código muito grandes. Notavelmente, o novo compilador C++ Clang tem sido um binário de 64 bits desde seu lançamento no RAD Studio 12.0.





What's New in <RAD Studio 12.2 Athens>



Ali Dehbansiahkarbon
Senior software developer

Loving the new Focus Mode in RAD Studio 12.2! 🎯 As a longtime Delphi developer, it minimizes distractions and boosts my productivity. Thanks, Embarcadero team!

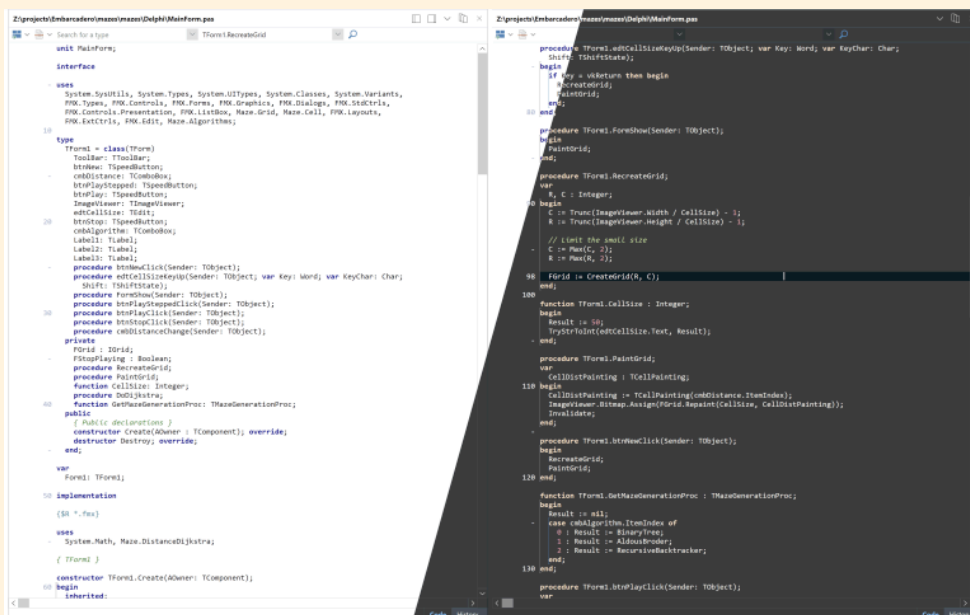


SEE WHAT'S NEW

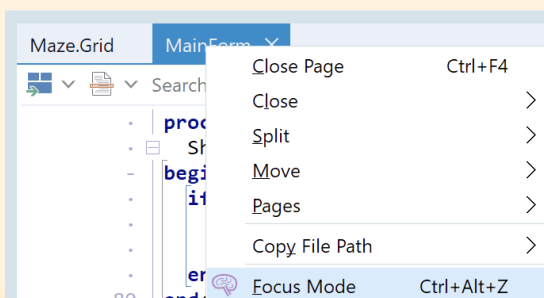
3 CÓDIGO MUITO MAIS RÁPIDO COM O MODO DE FOCO E ANOTAÇÕES NA BARRA LATERAL

O RAD Studio 12.2 apresenta o Focus Mode, um recurso de produtividade que deixa o IDE apenas com o(s) editor(es), eliminando todas as distrações e permitindo uma codificação mais rápida em um layout de IDE personalizável.

Enquanto isso, as novas anotações da barra de rolagem mostram rapidamente as seções do código que foram modificadas, têm marcadores, têm erros ou avisos e muito mais.



Para acessar o modo de foco, pressione Ctrl+Alt+Z ou clique com o botão direito do mouse na guia do editor e clique em Modo de foco.

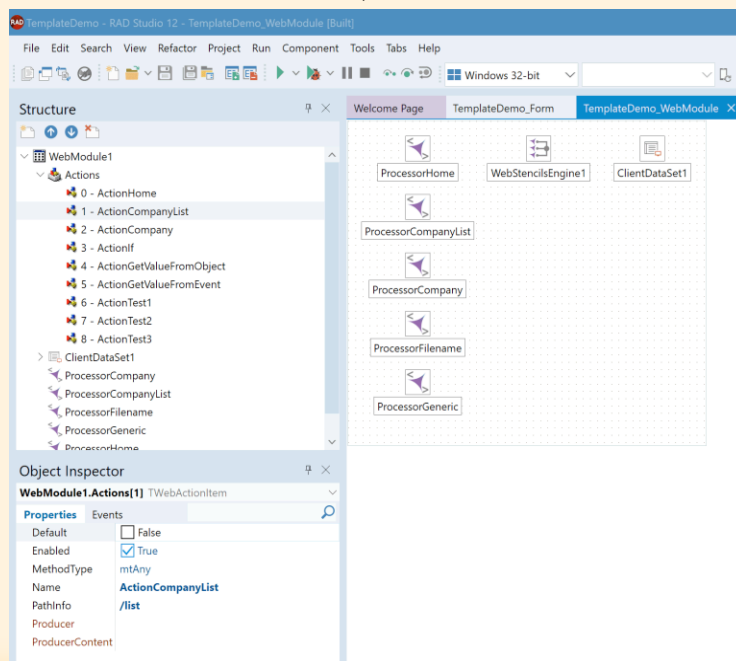




TURBINE O DESENVOLVIMENTO DA WEB COM O MECANISMO DE MODELOS DO WEBSTENCILS
 O RAD Studio 12.2 apresenta o WebStencils, uma integração e processamento de arquivos HTML com base em scripts do lado do servidor, também conhecido como mecanismo de modelo. Ele e seu componente podem ser usados no WebBroker e no RAD Server, mas também no DataSnap e até mesmo em aplicativos não Web para produzir arquivos HTML, JSON, YAML, delimitados por vírgula ou qualquer outro arquivo para mesclar dados específicos. O código do aplicativo pode ser escrito em **Delphi** ou em C++.

O principal objetivo do WebStencils é ampliar significativamente as tecnologias da Web existentes no RAD Studio (WebBroker, DataSnap, RAD Server), fornecendo scripts no lado do servidor e transformando o RAD Server de um mecanismo de serviço da Web em uma ferramenta de site e serviço da Web. Os usuários podem criar páginas HTML com ferramentas padrão e adotar qualquer biblioteca CSS e JavaScript, além de manter a capacidade de adicionar dados das páginas geradas a partir do aplicativo.

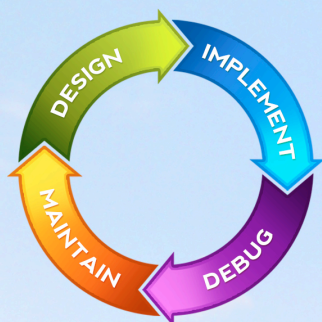
Além disso, o WebStencils pode ser uma boa base para o HTMX como uma solução para o desenvolvimento da Web. As duas tecnologias combinam muito bem. As páginas HTMX podem se beneficiar da geração de código no lado do servidor e se conectar a servidores REST com relação a atualizações de conteúdo. As tecnologias da Web do **Delphi** podem oferecer geração de páginas e APIs REST em um nível de alta qualidade.



Database Workbench 6.5

Now with **SQLite** support

the Swiss army knife for database development



Consistent user interface, modern code editors, Unicode enabled, HighDPI aware, ER designer, reverse engineering, meta data browsing, visual object editors, meta data migration, meta data compare, stored routine debugging, SQL plan visualizer, test data generator, meta data printing, data import and export, data pump, Grant Manager, DBA tasks, code snippets, SQL Insight, built in VCS, report editor, database meta data search, numerous productivity tools and much more...

for SQL Server, Oracle, MySQL, MariaDB, Firebird, InterBase, NexusDB and PostgreSQL



A próxima versão do kbmMW conterá um novo recurso que permite tornar declarações já curtas ainda mais curtas.

O recurso foi implementado com base em uma pergunta em uma das mídias sociais, na qual uma pessoa gostaria de obter a média de uma série de números incorporados em uma cadeia, cada número separado por um ponto e vírgula ';':

Imediatamente pensei em LINQ e escrevi um pequeno exemplo para o usuário, que tem a seguinte aparência:

AMOSTRA RESUMIDA (CÓDIGO)

```
procedure TForm1.Button1Click(Sender: TObject);
var
    sl:TStringList;
    ls:IkbmMWLinQStage;
begin
    sl:=TStringList.Create('',';');
    try
        sl.DelimitedText:='100.65;200.90;300.10';
        ls:=linq.Using(sl,[mwlusoNumericValue]).Shared;
        ShowMessage(Format('Min:%f Max:%f Avg:%f',
            [single(ls.Min),single(ls.Max),ls.Avg]));
    finally
        sl.Free;
    end;
end;
```

Como você pode ver, eu uso uma TStringList para carregar a string que contém os números. Como o kbmMW LINQ pode trabalhar com uma TStringList, foi simples selecionar os valores mínimo, máximo e médio.

No entanto, me incomodou o fato de ter que usar uma TStringList para fazer a divisão dos dados. Eu poderia ter usado um streamer de notação de objeto TkbmMWCSV, pois os dados são essencialmente valores separados por "vírgulas" (*exceto pelo uso de ponto e vírgula em vez de vírgula*).

Mas a escrita não seria mais curta, pois eu ainda precisaria criar uma instância do TkbmMWCSV, configurar uma ou duas propriedades e carregar os dados, antes de poder fornecer o objeto de notação carregado ao LINQ para fazer os cálculos.

De fato, o LINQ oferece suporte a instruções UsingCSV para torná-lo um carregamento de linha única, mas, infelizmente, ele só oferece suporte a conteúdos CSV que usam vírgulas como separadores, pois não permite nenhuma configuração do processo de carregamento dentro da própria instrução LINQ.

Como as instruções LINQ Usingxxxx já contém vários argumentos, e vários analisadores de dados têm muitas configurações possíveis disponíveis, não me senti à vontade para poluir as instruções Usingxxxx com argumentos adicionais, o que também entraria em conflito com as diversas variantes sobrecarregadas das instruções Using.

Portanto, foi necessário repensar um pouco, o que, inspirado pelo uso em outras partes do kbmMW, acabou ficando assim:

SHORTENED → SHORTHAND SAMPLE





SHORTENED → SHORTHAND SAMPLE

```

procedure TForm1.Button1Click(Sender: TObject);
var
    ls:IkbmMWLinQStage;
begin
    ls:=linq.WithSettings('recordDelimiterChar1:";"')
                .UsingCSV('100.65;200.90;300.10').Shared;
    ShowMessage(Format('Min:%f Max:%f Avg:%f',
        [single(ls.Min),single(ls.Max),ls.Avg]));
end;

```

A introdução da palavra-chave **LINQ** `WithSettings`. Ela permite que as configurações sejam fornecidas como uma cadeia de caracteres formatada em YAML simplificada, da mesma forma que as configurações são fornecidas em outros lugares no kbmMW, por exemplo, nos atributos REST e em outros lugares.

No exemplo acima, eu simplesmente uso o `UsingCSV`, mas somente depois de já ter especificado que quero que o caractere delimitador de registro seja um ponto e vírgula. Mais configurações podem ser fornecidas separando cada **propertyName: property-value** por uma vírgula.

Um efeito colateral interessante disso é que a string de configuração suporta a coleta das configurações de um centro de configuração (*Consulte Configuração e Configuração 2*).

Portanto, é possível escrever:

```

procedure TForm1.Button1Click(Sender: TObject);
var
    ls:IkbmMWLinQStage;
begin
    ls:=linq.WithSettings('recordDelimiterChar1:"$(delimiters.myRecordDelimiter=;)"')
                .UsingCSV('100.65;200.90;300.10').Shared;
    ShowMessage(Format('Min:%f Max:%f Avg:%f',
        [single(ls.Min),single(ls.Max),ls.Avg]));
end;

```

Nesse caso, ele primeiro procurará as configurações no centro de configuração no caminho `delimiters.myRecordDelimiter`. Se nada for encontrado lá, o padrão será usar um delimitador ; (*ponto e vírgula*).

Não sei com que frequência alguém usaria as configurações baseadas em configuração do LINQ, mas presumo que elas possam ser usadas, o que facilita muito a reconfiguração do aplicativo para variações de uma situação, bastando alterar um arquivo INI, uma configuração de registro, um arquivo **JSON**, **XML** ou **YAML** ou um banco de dados, seja qual for o tipo de armazenamento de configuração usado.





The following lists the more important settings available for some of the typical **Usingxxx** variants:

header	true/false (default false)
recordDelimiterChar1	record delimiter. Default \r (CR)
recordDelimiterChar2	opcional 2. caractere delimitador de registro. Default \n (LF)
fieldDelimiterChar	field delimiter. Default , (virgula)
quoteChar	string quoting character. Default " (aspas)
autoDetectType	true/false (default true). Se for verdadeiro, tentará detectar o tipo de dado com base no conteúdo de todos os dados lidos para esse campo específico.
defaultNamePrefix	prefixo para nomes de campos se lidos sem cabeçalho. Default FLD.
skip	0..n, default 0. Quantos registros, após o cabeçalho opcional, devem ser ignorados antes da leitura dos dados.
skipBeforeHeader	0..n, default 0. Quantas linhas de texto devem ser ignoradas antes do cabeçalho ou dos dados opcionais..
dontExceptOnEmptyString	true/false, default false. Se for fornecida uma cadeia de caracteres vazia para analisar, ela deve excluir ou retornar um conjunto de dados vazio?
limit	-1..n, default -1. Limite do número de registros de dados a serem lidos.
trim	true/false, default false. Se for para remover espaços prefixados ou pós-fixados dos dados da coluna, caso não estejam entre aspas.
skipNoNameColumns	true/false, default false. Se for para ignorar colunas que tenham um nome vazio no cabeçalho.
UsingCSV	
jQueryEscaping	true/false, default false. Determina se há suporte para dados JSON com escape do jQuery.
lateTyping	true/false, default false. Determina se os dados aparentemente numéricos devem ser lidos como uma cadeia de caracteres.
dontExceptOnEmptyString	true/false, default false. Se for fornecida uma cadeia de caracteres vazia para analisar, ela deve excluir ou retornar um conjunto de dados vazio.

Ele pode usar JSON e JSON5



Object Inspector

Components (filter)

Properties (filter)

Properties Events Favorites Re

```

1 { Directory cleaning component
2
3 Copyright (C) 2007 Michael Van Canneyt
4
5 This library is free software; you
6 under the terms of the GNU Library
7 the Free Software Foundation; either
8 option) any later version with the
9
10 As a special exception, the copyright holders of this lib
11 permission to link this library with independent modules
12 executable, regardless of the license terms of these ind
13 to copy and distribute the resulting executable under te
14 provided that you also meet, for each linked independent
15 conditions of the license of that module. An independ
16 which is not derived from or based on this library,
17 you may extend this exception to your vers
18 not obligated to do so. If you do not wish to
19 statement from your version.
20
21 This program is distributed in the hope that it will be
22 useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCH
23 FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library Ge
24 for more details.
25
26 You should have received a copy of the GNU Library Gener
27 along with this library; if not, write to the Free Softw
28 Inc., 51 Franklin Street - Fifth Floor, Boston, MA 02110
29
30 unit dircleaner;
31
32 {$mode objfpc}{$SH+}
33
34 interface
35 uses
36 Classes, SysUtils, iniicol, IniFiles, LazFileUtils, LazUT
37
38 type
39 TSubDir = (sdExtension, sdYear, sdMonth, sdDay, sdDate, sdHour);
40 TSubDirs = Set of TSubDir;
41
42 { TLocation }
43
44 TLocation = Class(TNamedIniCollectionItem)
45 private
46 FBasePath: String;
47 FSubDirs: TSubDirs;
48 procedure SetSubDirs(const AValue: TSubDirs);
49 Public
50 Function GetLocation(AExtension: String; ATime: TDate
51 Procedure AddLocation(AExtension: String; ATime: TDate
52 Procedure LoadFromIni(Ini: TCustomInifile; Section: Str
53 Procedure LoadFromIni(Ini: TCustomInifile; Section: S
54 Published
55 Property BasePath: String;
56 Property SubDirs: TSubDirs;
57 end;
58
59 { TLocation }
60
61 TLocation = Class(TNamedIniCollectionItem)
62 private
63 FBasePath: String;
64 FSubDirs: TSubDirs;
65 FLocationName: String;
66 FMinCompressSize: Integer;
67 FWorkExt: String;
68 Public
69 Constructor Create(ABasePath: String; ASubDirs: TSubDir
70 Function AddLocation(AExtension: String; ATime: TDate
71 Function FileNameFits(AFileName: String): Boolean;
72 Function Property Location: TLocation;
73 Property LocationName: String;
74 end;
75
76 TFileAction = Class(TNamedIniCollectionItem)
77 private
78 FCaseSensitive: Boolean;
79 FCompress: Boolean;
80 FDelete: Boolean;
81 FExtension: String;
82 FLocation: TLocation;
83 FLocationName: String;
84 FMinCompressSize: Integer;
85 FWorkExt: String;
86 Public
87 Procedure Prepare;
88 Function FileNameFits(AFileName: String): Boolean;
89 Property Location: TLocation;
90 Procedure SaveToIni(Ini: TCustomInifile; Section: Str
91 Procedure LoadFromIni(Ini: TCustomInifile; Section: S
92 Procedure Assign(Source: TPersistent); override;
93 Published
94 Property LocationName: String;

```

IDE Scout (527)

Type search terms

Instant search

Enter search text, at least 3 characters

File Browser

Configure Reload

All Files (*)

- CA
- DA
- EA
- FA
- GA
- HA
- JA
- C:\Lazarus\examples\cleandir

All Files (*)

DumpStack.log

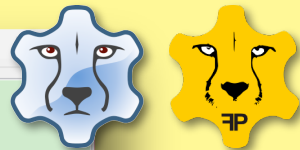
Recovery.txt

Se você visualizar a versão em PDF, poderá ver essas páginas lado a lado para ter uma visão geral melhor Visualização de codificação normal

MiniMap para escanear rapidamente a parte do código do documento

Visualização de codificação normal

Lazarus Trunk Version 4.0



O complemento IDE Scout mostra o "IDE Scout", uma janela de pesquisa rápida para comandos do IDE. Ele funciona como a paleta de comandos no Atom ou no Visual Studio Code.

IA Inteligência Artificial:
Digite aqui seu pedido ou solicitação

Pesquisa instantânea para escanear rapidamente a parte do código do documento

Navegador de arquivos para escanear rapidamente a parte do código do documento

Mensagem para escanear rapidamente a parte do código do documento

Toggle Form/Unit View	F12
Object Inspector	F11
Source Editor	
Messages	
Code Explorer	
FPCDoc Editor	
Code Browser	
Unit Dependencies	
Restriction Browser	
Components	Ctrl+Alt+P
Jump History	Ctrl+Alt+J
Editor Macros	
Test Insight	
File Browser	
File Searcher	Ctrl+P
Instant search	Ctrl+Alt+F
Project Groups	
ToDo List	
Assist chat	Ctrl+Alt+I, Ctrl+Alt+C
Simple Web Server	
Leaks and Traces	
Anchor Editor	
Tab Order	
Search Results	Ctrl+Alt+F
Debug Windows	>
IDE Internals	>

Toggle Form/Unit View	F12
Object Inspector	F11
Source Editor	
Messages	
Code Explorer	
FPCDoc Editor	
Code Browser	
Unit Dependencies	
Restriction Browser	
Components	Ctrl+Alt+P
Test Insight	
File Browser	
File Searcher	Ctrl+P
Instant search	Ctrl+Alt+F
Project Groups	
ToDo List	
Assist chat	Ctrl+Alt+I, Ctrl+Alt+C
Simple Web Server	
Leaks and Traces	
Anchor Editor	
Tab Order	
Search Results	Ctrl+Alt+F
Debug Windows	>
IDE Internals	>

About FPC
About IDE
What Needs Building
Debug Output
Show IDE Scout ... Shift+Alt+P





The image displays two screenshots of the Lazarus Source Editor showing the Delphi code for the 'dircleaner' component. The code is organized into sections, with red arrows pointing from the lion heads to specific parts:

- Left Screenshot:** Shows the top part of the code, including the 'dircleaner' component declaration, a 'uses' section, and the 'TLocation' type definition. A red arrow points to the copyright notice.
- Middle Screenshot:** Shows the 'TLocation' type definition, the 'TLocations' class declaration, and the 'TLocations' class implementation. A red arrow points to the 'SaveToIni' procedure.
- Right Screenshot:** Shows the 'TFileAction' class declaration and implementation. A red arrow points to the 'TFileAction' class implementation.

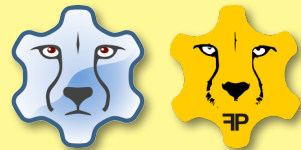
```
1 { Directory cleaning component
2
3 Copyright (C) 2007 Michael Van Canneyt
4
5 This library is free software; you can redistribute it a
6 under the terms of the GNU Library General Public Licen
7 the Free Software Foundation: either version 2 of the Li
8 (option) any later version with the following modificati
9
10 As a special exception, the copyright holders of this lib
11 permission to link this library with independent modules
12 executable, regardless of the license terms of these ind
13 to copy and distribute the resulting executable under te
14 provided that you also meet, for each linked independe
15 and complete the corresponding copyright notice for indepen
16 module which is not derived from or based on this libra
17 this library, you may extend this exception to your vers
18 but you are not obligated to do so. If you do not wish t
19 exception statement from your version.
20
21 This program
22 ANY WARRANTY
23 FITNESS FOR
24 FOR MORE DET
25
26 You should ha
27 along with th
28 Inc., 51 Fra
29
30 unit dircleaner;
31 ($mode objfpc);
32
33 interface
34
35 uses
36   Classes, Sys
37
38 type
39   TSubDir = se
40   TSubDirs = S
41
42 { TLocation
43
44 TLocation = c
45
46 private
47   FBasePath:
48   FSubDirs: T
49   procedure
50   Public
51   Function Ge
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```





The screenshot displays the Lazarus IDE interface. On the left, the File Browser window shows a directory tree for 'C:\lazarus\examples\cleandir' with a 'backup' folder selected. A red arrow points from the 'backup' folder to the 'File browser options' item in the IDE Options dialog. The IDE Options dialog is open, showing the 'Environment' section with 'File browser options' selected. A context menu is open over the IDE Options dialog, listing various IDE components. A red arrow points from the 'File Browser' option in the context menu to the File Browser window. A yellow callout box with a speech bubble contains the text: 'Navegador de arquivos para escanear rapidamente a parte do código do documento'.





IDE Scout (509/549)

T|

- code tools: add a "with" block
- codetools: add assign method (Alt+F11)
- codetools: add unit to uses section (Ctrl+J)
- codetools: check lfm file in editor (Ctrl+Shift+C)
- codetools: code template completion (Ctrl+Shift+X)
- codetools: complete code
- codetools: complete code (with dialog)
- codetools: declare variable
- codetools: explode a "with" block
- codetools: extract procedure
- codetools: find block other end (Ctrl+Q, O)
- codetools: find block start (Ctrl+Q, M)
- codetools: find declaration (Alt+Up)
- codetools: find identifier references (Ctrl+Shift+I)
- codetools: find procedure definition (Ctrl+Shift+Up)
- codetools: find procedure method (Ctrl+Shift+Down)
- codetools: find procedure/method overloads
- codetools: find references of used unit
- codetools: goto include directive
- codetools: guess misplaced Sifdef
- codetools: guess unclosed block
- codetools: identifier completion (Ctrl+Space)
- codetools: insert call inherited
- codetools: insert int64 id yyyyddmmhhnss
- codetools: invert assignment
- codetools: jump to
- codetools: jump to implementation
- codetools: jump to implementation uses
- codetools: jump to initialization
- codetools: jump to interface
- codetools: jump to interface uses

Complemento IDE Scout mostrando o "IDE Scout" - uma janela de pesquisa rápida para comandos do IDE.

Digite um comando ou nome de tipo como TButton e ele criará uma lista e você poderá clicar no texto com mais informações.

- Toggle Form/Unit View F12
- Object Inspector F11
- Source Editor
- Messages
- Code Explorer
- FPDoc Editor
- Code Browser
- Unit Dependencies
- Restriction Browser
- Components Ctrl+Alt+P
- Jump History Ctrl+Alt+J
- Editor Macros
- Test Insight
- File Browser
- File Searcher Ctrl+P
- Instant search Ctrl+Alt+F
- Project Groups
- ToDo List
- Allssist chat Ctrl+Alt+I, Ctrl+Alt+C
- Simple Web Server
- Leaks and Traces
- Anchor Editor
- Tab Order
- Search Results Ctrl+Alt+F
- Debug Windows
- IDE Internals

IDE Scout

Find option [Ctrl+F]

Environment

- Files
- General
- Window
- IDE CoolBar
- Editor ToolBar
- Component Palette
- Form Editor
- Object Inspector
- Messages Window
- FPDoc Editor
- IDE Startup
- Backup
- Naming
- File Filters
- Docking / Anchordocking
- Test Insight
- Project Groups
- Docked Form Editor
- Simple Web Server
- Pas2JS
- File browser options
- IDE Scout
- Instant search
- AI Assistant options

Search scope

- Commands
- Recent Projects
- Recent Files
- Recent Packages
- Components

Options

- Show Shortcut when available
- Show Category when available

Colors

Matches: Shortcut:

Components

- Only select on component palette

Default width 0 Default height 0

Help Apply OK Cancel

- About FPC
- About IDE
- What Needs Building
- Debug Output
- Show IDE Scout ... Shift+Alt+P





INTERNATIONAL FREE PASCAL & LAZARUS MEETING UNIVERSITY COLOGNE

THURSDAY 10 AND FRIDAY 11 OCTOBER 2024
<https://www.blaisepascalmagazine.eu/en/pascalcafe2024>



UNIVERSITÄT ZU KÖLN

Regionales Rechenzentrum (RRZK)



BLAISE PASCAL  MAGAZINE





FPC/^{Lazarus} FRESNEL

Now fpc/^{Lazarus} using fresnel
Has three working backends,
A css-driven layout,
Multiple platforms,
A powerful event mechanism.

We now can:

CREATE A UNIVERSAL GRAPHICAL APPLICATION RUNNING ON ALL NATIVE PLATFORMS AND IN THE BROWSER.

All this using a single codebase,
and running at native speed.
And obviously,
all this using your favourite Programming language:

OBJECT PASCAL.



Graças a Claude Shannon



INTRODUÇÃO

Sempre ouvimos falar que a IA e os modelos de linguagem grandes tomarão todos os nossos empregos. Em 1980, quando eu soube que queria ser programador de computadores, disseram-me que os computadores se programariam sozinhos em breve e que eu ficaria sem emprego.

Isso ainda não aconteceu, mas a tecnologia mais recente certamente está mudando nosso trabalho mais do que em qualquer outro momento no passado. É um momento muito interessante (tanto no bom quanto no mau sentido) para trabalhar com software. Então, como é exatamente escrever código **Delphi** com um assistente de modelo de **linguagem grande**?

O QUE É UM MODELO DE LINGUAGEM GRANDE?

Fornecerei apenas algumas noções básicas, já que isso está mudando muito rapidamente. Também apresentarei alguns antecedentes históricos muito interessantes.

Se você é completamente novo em **LLM** e **GPT**, saiba que posso usar esses termos de forma intercambiável, mas, tecnicamente, um **Generative Pre-trained Transformer (GPT)** é um tipo de **Large Language Model (LLM)**. Eles se enquadram na categoria de **Inteligência Artificial Generativa** criada a partir de **Redes Neurais Artificiais** que são treinadas em tarefas de **Processamento de Linguagem Natural**. [*Siga os links da Wikipédia para obter informações atualizadas mais recentemente*].

COMPREENSÃO DAS REDES NEURAIIS ARTIFICIAIS

A ideia de uma rede neural artificial existe há muito tempo, mas foi apenas recentemente, graças à natureza paralela maciça das GPUs modernas, que ela se tornou prática.

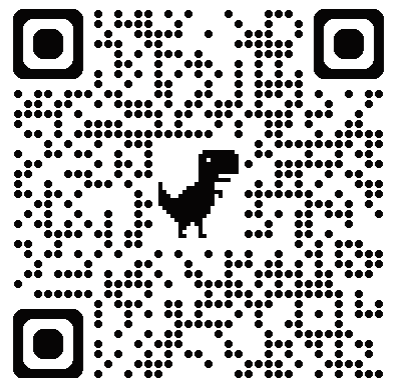
Essencialmente, eles são um modelo matemático baseado na rede neural em um cérebro biológico que é treinado em grandes quantidades de dados de entrada.

Isso faz com que a estrutura interna da rede neural seja uma "caixa preta", em que o comportamento é resultado do treinamento e não é especificado diretamente pelo código normal. Embora sejam treinadas com grandes quantidades de dados, elas não armazenam os dados com os quais foram treinadas. Em vez disso, elas retêm os padrões dos dados, o que lhes permite gerar um novo texto que corresponda ao mesmo padrão.

3Blue1Brown acaba de publicar um vídeo fantástico sobre o GPT especificamente. Se você não estiver familiarizado com **3Blue1Brown**, é um canal incrível de educação matemática no YouTube, criado por Grant Sanderson. Ele é um dos melhores educadores que já encontrei sobre esses tópicos e tem mais de 170 vídeos, abrangendo várias listas de reprodução. O vídeo do GPT faz parte de um curso sobre redes neurais.

UM POUCO DE HISTÓRIA INTERESSANTE

Claude Shannon é uma pessoa sobre a qual poucas pessoas falam quando se trata de modelos de linguagem ampla. Não tenho certeza de onde aprendi sobre o **Sr. Shannon**, mas presumi que ele era uma daquelas pessoas como **Charles Babbage**, **Ada Lovelace**, **Blaise Pascal**, **Margaret Hamilton**, **George Boole**, **Hedy Lamarr**, **Alan Turing** e **Grace Hopper** que todo mundo conhecia (*inclui links por um motivo, vá ler sobre eles*). Mas, surpreendentemente, ele parece ser menos conhecido. Em 1950, ele construiu o **Theseus**, o primeiro exemplo de inteligência artificial, um dispositivo que aprendia por tentativa e erro. Ele também foi o pai da Teoria da Informação, lançou as bases para que a **álgebra booleana** fosse usada na computação digital e muito mais. Ele é descrito como o engenheiro do século XX que mais contribuiu para as tecnologias do século XXI. Diz-se que suas realizações estão no mesmo nível das de **Albert Einstein** e **Sir Isaac Newton** em seus campos.



Graças a Claude Shannon

QUEM É CLAUDE SHANNON?

Há um documentário fantástico sobre Claude Shannon chamado Bit Player que eu recomendo muito. (Também encontrei uma versão com legendas em português do Brasil).

Aqui está um documentário mais antigo, mas eu prefiro o Bit Player.

Também acabei de descobrir uma biografia sobre ele, **A Mind at Play: How Claude Shannon Invented the Information Age**, de **Jimmy Soni, Rob Goodman**, mas ainda não a li.

Eu ouvi um ótimo podcast com os autores.

Além de tudo isso, o que considero mais interessante e relevante para o **Large Language Models** é sua pesquisa sobre "pares de palavras".

A ideia é que, se você observar qual palavra normalmente segue outra palavra, poderá simular uma escrita "inteligente".

Como ele criou essa ideia antes da invenção dos computadores digitais, ele testava sua teoria com um romance, procurando a próxima ocorrência de uma palavra, copiando a palavra depois dela e repetindo o processo até chegar a uma frase crua.

Essa é uma versão muito simplificada do que um modelo de idioma grande faz.

Por isso, achei que seria interessante usar um modelo de linguagem grande e moderno para criar um modelo de linguagem pequeno e primitivo. Se você der uma olhada nos dados de análise salvos, verá que eles contêm apenas uma lista de palavras e números, mas não contêm o texto original analisado.



USANDO O CHATGPT 4 PARA ESCREVER CÓDIGO Delphi

Na medida do possível, usei o ChatGPT 4 para escrever o código. Fiz algumas pequenas edições por conta própria e criei o aplicativo de amostra, mas o ChatGPT fez a maior parte do código da biblioteca e todos os testes de unidade. Compartilhei o histórico da minha conversa com o ChatGPT para que você possa ver o processo. Ele se saiu muito bem dessa vez.

- [ChatGPT 4 conversation](https://chat.openai.com/share/4e41c575-c1cc-4fd2-975f-f4f46aa522b7)

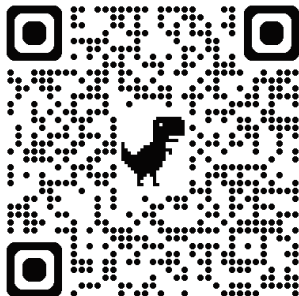
<https://chat.openai.com/share/4e41c575-c1cc-4fd2-975f-f4f46aa522b7>

- [Conversation with comments](https://docs.google.com/document/d/e/2PACX-1vSuB7gXH9BO_IN5MgJv31cImVdxZ6jNtw3kqXG4Gc4GydNNVzHqMf7e27cYFG_wrNDqmLWzCqVTE09R/pub)

https://docs.google.com/document/d/e/2PACX-1vSuB7gXH9BO_IN5MgJv31cImVdxZ6jNtw3kqXG4Gc4GydNNVzHqMf7e27cYFG_wrNDqmLWzCqVTE09R/pub

- [Full Source Code on GitHub](https://github.com/jimmckeeth/Delphi-Shannon-Word-Pairs)

<https://github.com/jimmckeeth/Delphi-Shannon-Word-Pairs>



Graças a Claude Shannon

O QUE APRENDEMOS?

Não cronometrei o tempo, mas acredito que foi mais rápido do que se eu mesmo tivesse escrito todo o código. O modelo de linguagem grande realmente se destaca quando você está trabalhando em algo com o qual não está tão familiarizado, evitando a necessidade de consultar a documentação o tempo todo. Pode ser um algoritmo com o qual você não está familiarizado, uma API com a qual você não está familiarizado, etc. Se você tentar usar o ChatGPT ou um modelo de linguagem grande semelhante para escrever código, aqui estão algumas coisas com as quais você deve se preocupar:

- **Alucinações:**

Eu tinha um amigo que respondia com confiança a qualquer pergunta que você fizesse a ele, mesmo que ele não tivesse ideia do que estava falando. Isso é exatamente o mesmo que um LLM. A menos que você peça a ele para fazer algo que foi especificamente treinado para não fazer, ele lhe dará uma resposta e agirá com muita confiança.



ChatGPT



Google



Microsoft

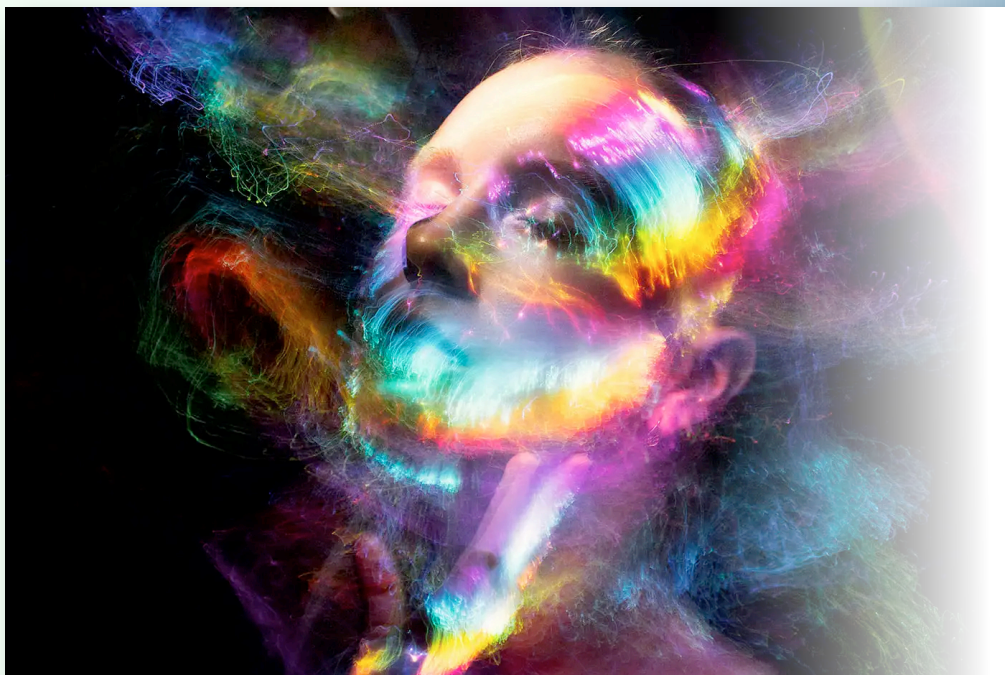


Ilustração do ChatGPT - O que são alucinações e por que elas são um problema para os sistemas de IA - [Bernard Marr](https://bernardmarr.com/chatgpt-what-are-hallucinations-and-why-are-they-a-problem-for-ai-systems/), <https://bernardmarr.com/chatgpt-what-are-hallucinations-and-why-are-they-a-problem-for-ai-systems/>

- **Você precisa saber como codificar:**
Como ele pode lhe dar lixo, você precisa aplicar seu conhecimento para ver se o resultado faz sentido e se está cumprindo o que você pediu.
- **Iterar:**
Faça com que o LLM realize pequenas quantidades de trabalho e dê feedback à medida que você avança, fazendo alterações.
- **Aprendizado Zero-Shot:**
Se ele estiver completamente perdido, forneça um código de exemplo. Às vezes, isso é suficiente para que ele siga na direção certa.
- **Dados de treinamento mais antigos:**
A maioria dos LLMs é treinada com dados antigos, o que significa que nem sempre estão atualizados com as últimas alterações na **Delphi**. Felizmente, o ChatGPT 4, o Bing CoPilot e outros podem ficar on-line e se atualizar. Em minha experiência, isso produz resultados mistos, mas se, por exemplo, você pedir que ele aproveite as declarações de variáveis em linha, ele geralmente descobrirá.



Graças a Claude Shannon



- **Escopo limitado/janela de token:**

Basicamente, se você tiver um bate-papo muito longo, às vezes ele "esquecerá" coisas que você discutiu ou perguntou no início. Isso ocorre porque cada etapa do bate-papo inclui apenas uma parte limitada da conversa anterior. A duração dessa retrospectiva está ficando cada vez mais longa, mas ainda assim vale a pena ter em mente.

- **LLMs não são mecanismos de busca:**

Se você observar o resultado da análise do programa **Delphi**, verá que ele não contém o texto de entrada que foi analisado. Isso é o mesmo que um LLM.

Isso significa que ele não "copiou" o trabalho original e também significa que provavelmente não o reproduzirá exatamente.

É totalmente possível que ele a reproduza, mas também é possível que um milhão de macacos em um milhão de máquinas de escrever escrevam Shakespeare.

- **Dados limitados do treinamento Delphi:**

Embora eu não saiba ao certo em que esses **LLMs** públicos são treinados, tenho certeza de que foram treinados no DocWiki e em grande parte do código **Delphi** no GitHub e em outros lugares da Web.

Provavelmente não foi treinado na **RTL** em si, mas provavelmente foi treinado na **RTL** do Free Pascal (por ser de código aberto e estar disponível gratuitamente on-line.)

O resultado é que tenho muitas alucinações em que ele afirma com confiança que um método ou classe existe, quando não existe.

Novamente, isso se deve ao fato de não ser um mecanismo de pesquisa.

Ele não procura o método, mas extrapola o que espera que o método seja chamado e como ele funciona.

Às vezes, ele usa um método FPC, outras vezes usa um método .NET.

Normalmente, você pode fornecer feedback e ele se corrigirá sozinho, mas houve algumas vezes em que ele insistiu que eu estava errado.

- **Engenharia imediata:**

Essa é uma forma de arte totalmente nova sobre como fazer perguntas aos **LLMs** que produzam as respostas que você está procurando. Você pode ler sobre o que os outros estão fazendo ou pode simplesmente experimentar. Se não estiver obtendo uma boa resposta, tente perguntar de outra forma. É interessante notar que, às vezes, oferecer uma recompensa ao **LLM faz** com que ele se torne mais criativo ou produza resultados diferentes.

- **Sincronização de alterações:**

Uma coisa que me atrapalhou algumas vezes foi quando fiz alterações no código depois de recebê-lo do **ChatGPT** sem informar ao **ChatGPT** sobre a alteração. Então, quando eu pedia mais código, ele não sabia das minhas alterações e, por isso, a saída não estava sincronizada com o meu código. Por exemplo, eu renomeei um método, portanto, toda vez que ele gerava um novo código que chamava esse método, eu precisava renomeá-lo novamente.

A solução seria inserir meu código alterado de volta na conversa.

O QUE VEM A SEGUIR?

Há opções gratuitas para usar o **Bing CoPilot**, o **ChatGPT 3.5**, o **Google Gemini** e outros.

Se você ainda não tentou trabalhar com eles, realmente precisa reservar algum tempo para se familiarizar. Se você já tentou há algum tempo, tente novamente.

As coisas estão mudando muito rapidamente neste momento e, embora a IA possa não tomar diretamente o seu emprego, se os seus colegas de trabalho aprenderem a usar esses LLMs para serem mais produtivos, eles poderão torná-lo redundante. Tudo isso é demais para você?

Bem, você está com sorte! A **GDK Software** está totalmente focada no mundo em rápida evolução do desenvolvimento de software **Delphi**. Tudo, desde a atualização de seu antigo projeto **Delphi 7** até a implementação dos mais recentes recursos e funcionalidades de aprendizado de máquina, está sob nossa responsabilidade.



ADVERTISEMENT

1. One year Subscription

2. Internet Viewing of the Magazine

3. The newest LIB Stick

BLAISE PASCAL MAGAZINE 118/119

Articles:

- From the editor
- Magnum, the new solution for QuartusII
- Video Effects and Animations creating video effect without hardly any coding
- Different Kind of Logic / Sociates
- FreePascal - Report - Part Two A new ReportingFrame for LAZARUS
- Working with TChart
- FreePascal - Report - Part Two A new ReportingFrame for LAZARUS

4. **Lazarus HANDBOOK** POCKET Edition + shipment

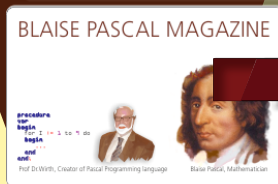
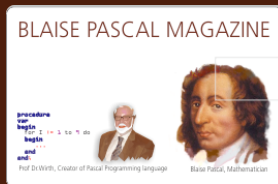
5. **Lazarus HANDBOOK** PDF

6. **LEARN TO PROGRAM USING LAZARUS** HOWARD PAGE-CLARK

7. **DAVID DIRKSE** including 50 example projects

BLAISE PASCAL MAGAZINE: **COMPUTER (GRAPHICS) MATH & GAMES IN PASCAL**

1. One year Subscription
2. Internet Viewing of the Magazine
3. The newest LIB Stick
 - All issues 1-111
 - On Credit Card
4. **Lazarus Handbook Pocket**
5. LH PDF including Code
6. Book Learn To Program
 - using **Lazarus PDF** including 19 lessons and projects
7. Book Computer Graphics Math & Games
 - PDF including ±50 projects

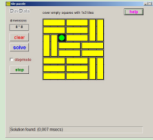


**SUPER
PACK**

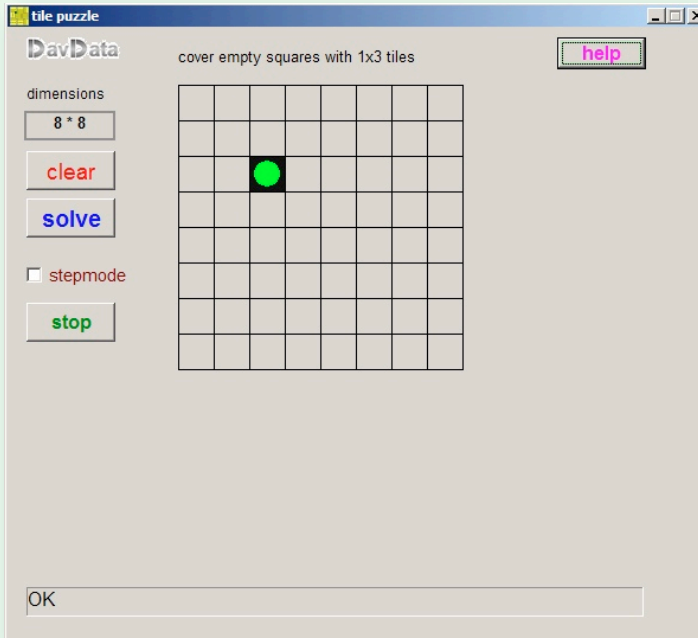
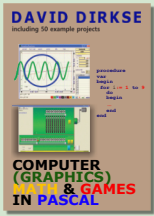
7 ITEMS

PRICE € 120

NORMAL PRICE € 275

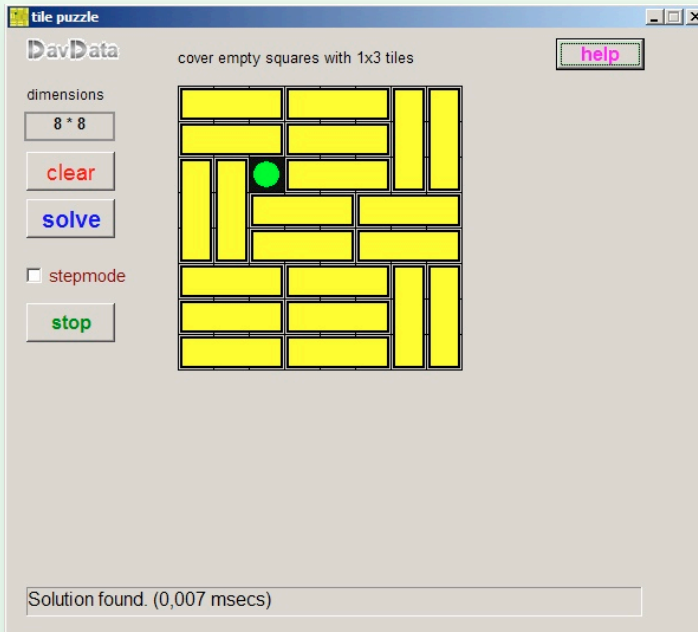


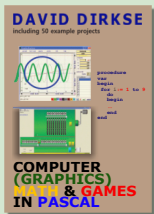
Delphi 12 Lazarus



Vemos um tabuleiro de 8x8 quadrados. Um quadrado é de cor verde. Os quadrados vazios devem ser cobertos por peças de 1x3.

Abaixo está a solução:





QUAIS DIMENSÕES PERMITEM UMA SOLUÇÃO?

Se as dimensões da placa forem $d \times d$, então $((d*d)-1) \bmod 3$ deve ser 0.

DIMENSÕES VARIÁVEIS:

Um clique com o botão esquerdo do mouse no botão de dimensões diminui o tamanho.

Um clique com o botão direito do mouse no botão de dimensões aumenta o tamanho.

As dimensões selecionáveis são 4x4, 5x5, 7x7, 8x8, 10x10, 11x11, 13x13

pele motivo acima.

Colocação do ladrilho verde 1x1:

No tabuleiro vazio: clicando com o mouse no quadrado.

Se houver azulejo verde: coloque o ponteiro do mouse sobre o bloco verde, abaixe o mouse e mantenha-o pressionado enquanto desloca o bloco para a nova posição.

Solte o botão do mouse se o bloco estiver no local correto.

COLOCAR UM AZULEJO AMARELO 1X3:

Mova o ponteiro do mouse sobre o quadro. Os pontos indicam posições livres de 1x3.

Passa o mouse para baixo e mova sobre os quadrados a ser coberto.

Remoção de um ladrilho amarelo: Clique com o mouse no bloco.

Botão Solve Pressione para iniciar a busca da solução no computador.

Stop button: Pressionar o botão de parada encerra a pesquisa no computador.

DESCRIÇÃO DO PROGRAMA

A programação é a manipulação de estruturas de dados.

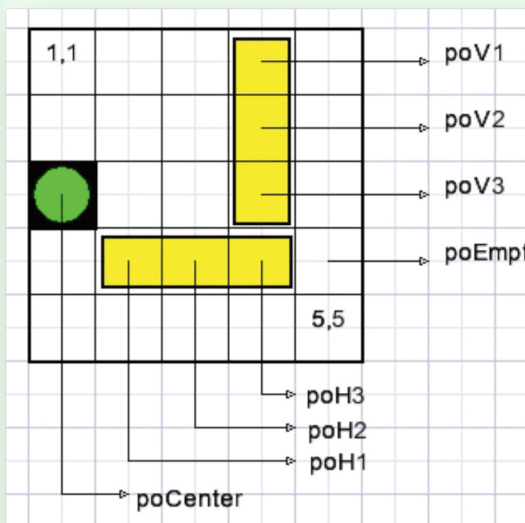
DIMENSÕES

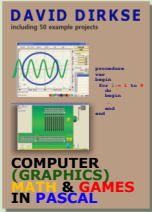
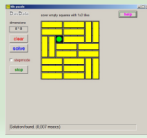
```
const dimensions : array[1..7] of byte = (4,5,7,8,10,11,13);
var dimcode : byte = 5; //1..7
    dim : byte = 10; //board size dim*dim
```

STATUS

Each board square has a status:

```
type Tpos3 = (poEmpty, poH1, poH2, poH3, poV1, poV2, poV3, poCenter);
var board : array[1..13,1..13] of Tpos3; //maximum size
```





POSIÇÕES DOS LADRILHOS:

```
type
  Ttile3 = record
    dir : byte; // 1:right 2:up 4:left 8:down
    posX, posY : byte; //board position of left/top square of tile
  end;
```

The Ttile3 **type** is used for drawing the yellow tiles.

```
var tile3 : Ttile3;
    greenX : byte = 5; //X position of green tile
    greenY : byte = 5; //Y...
```

A pesquisa no computador precisa de uma lista de movimentos.

```
type
  Tmove = record
    dir : char; // 'H' (horizontal or 'V'(vertical) tile
    mx : byte; //board coordinates
    my : byte; //..
  end;
var x,y : board // position during search
    searchBusy : boolean; //blocks unwanted events while searching
    pauseFlag : boolean; //control stepwise execution
    abortflag : boolean; //causes search end
    movecount : byte; //number of tiles added
    movelist : array[1..56] of TMove;
    maxmove : byte; //(dimension^2-1) / 3
```

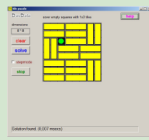
A função GoSearch lida com a busca de soluções no computador:

```
type TsearchResult = (srNo, srYes, srAbort);

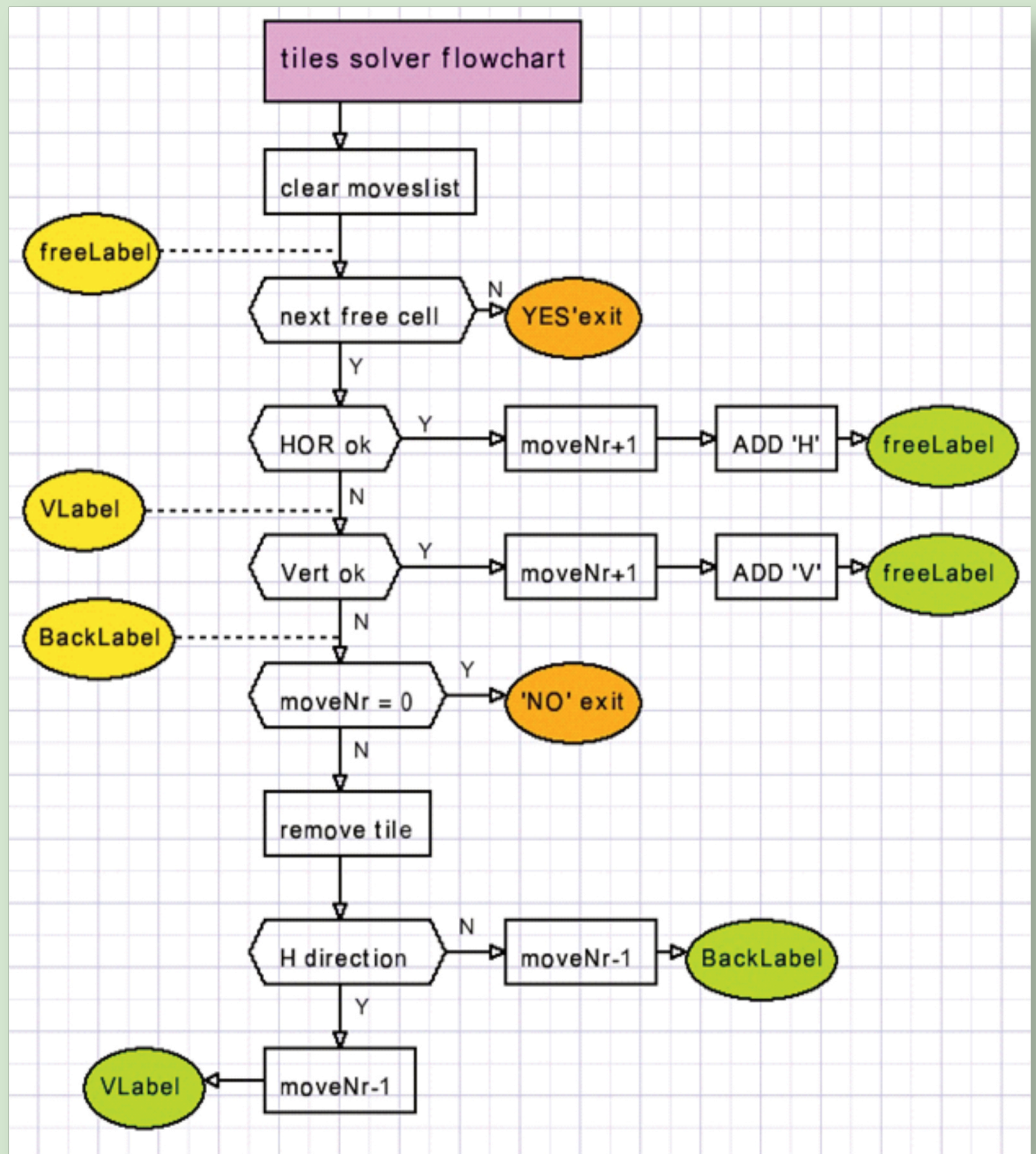
function GOsearch : TsearchResult;
var x,y : byte; //board position
    moveNr : byte;
    movelist : array[1..56] of TMove;
begin
  .....
End;
```

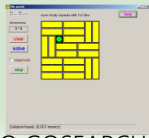
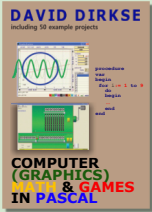
```
Searchresult srNO : no solution
                srYes : solution found
                srAbort : search ended by pressing the STOP button
```





FLUXOGRAMA DE PESQUISA NO COMPUTADOR





PROCEDIMENTOS NO GOSEARCH:

- **HorOK** : Retorna true se houver células livres (x,y) $(x+1,y)$ $(x+2,y)$
- **VertOk** : Retorna true se houver células livres (x,y) $(x,y+1)$ $(x,y+2)$
- **removeHV** : exclui o bloco para (x,y)
- **AddH** : adicione um ladrilho amarelo horizontal em (x,y)
- **AddV** : adicione um ladrilho amarelo vertical em (x,y)

A pesquisa começa com a limpeza da lista de movimentos.

Em seguida, o próximo quadrado vazio (célula) é procurado.

A pesquisa é feita da esquerda para a direita, de cima para baixo.

Nenhuma célula vazia encontrada significa que o quebra-cabeça foi resolvido.

Na próxima célula vazia, é feito um teste para um bloco horizontal de 1x3.

Se o teste falhar, tente um ladrilho vertical.

Se não houver espaço, o último bloco será removido.

Se esse ladrilho for horizontal, um ladrilho vertical será testado nessa posição.

Se o último bloco for do tipo "V", o bloco anterior será excluído.

Todas as peças removidas significam: não há solução.

Devido ao cruzamento de loops, as instruções GOTO e os rótulos são usados.

Isso resulta em um código mais legível, em que o uso de while..end ou repeat...until precisa de sinalizadores mágicos para lidar com os loops.

(O fluxograma apresenta rótulos em elipses coloridas)

Fato interessante:

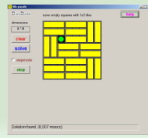
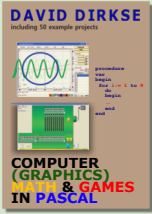
Parece que o retrocesso é supérfluo.

Até o momento, não encontrei uma solução que exigisse retrocesso.

Durante a pesquisa, um tempo limite variável faz a contagem regressiva de 1.000.000 até zero para cada bloco colocado.

Ao chegar a zero, o contador é redefinido para 1000.000, o quadro é exibido e o comando application.processmessages é executado para manter contato com o Windows.





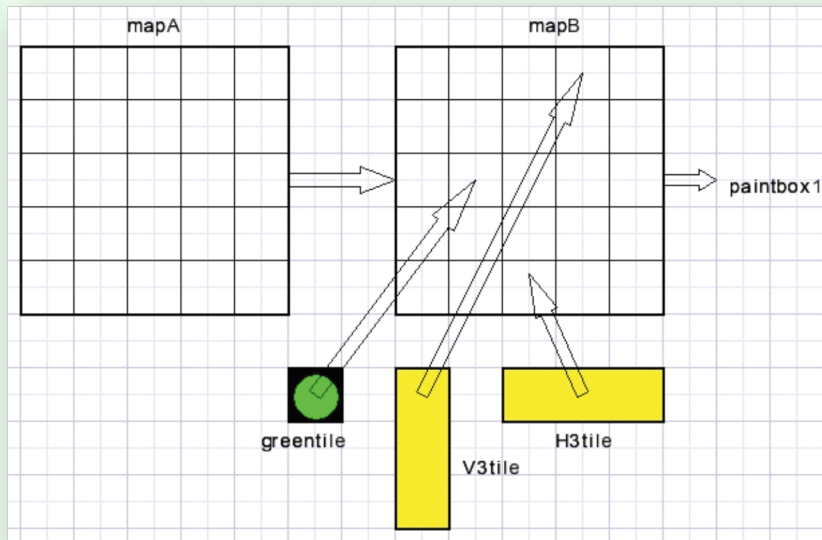
DESENHO.

Vários bitmaps são usados:

- **MapA** : mantém o quadro vazio. Usado para apagar o mapB
- **MapB**: segurar o tabuleiro + peças amarelas + peça verde
- **Greenmap** : O ladrilho de 1x1 contém o quadrado central verde
- **H3tile** : azulejo horizontal amarelo
- **V3tile** : azulejo vertical amarelo

Os mapas 1, 3, 4 e 5 são pintados no momento da criação.

Abaixo, os mapas são mostrados para um quadro de dimensão 5x5.



Um clique do mouse em um tabuleiro vazio coloca a peça verde.

Um clique do mouse sobre o bloco verde define a bandeira central.

Os movimentos subsequentes do mouse agora deslocam o bloco verde.

O movimento do mouse para cima limpa a bandeira central.

Um clique do mouse em um tabuleiro com uma peça verde coloca pontos nas células onde uma peça amarela pode ser colocada.

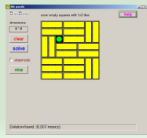
Os seguintes movimentos do mouse testam a direção.

Se duas células forem cobertas, o bloco amarelo será colocado no mapa B e o regflag será definido para impedir que a mesma operação ocorra novamente quando o ponteiro do mouse se mover.

Regflag é apagado por um clique do mouse.

Para apagar um bloco no mapa B, um `mapB.canvas.copyrect(r,mapA.canvas,r)` é usado, onde `r` é o retângulo necessário para copiar.





STEPMODE

A pesquisa no computador pode ser executada passo a passo.

Quando a caixa de seleção steppmode é marcada, após cada bloco colocado, o quadro é copiado para o paintbox1 e é feita uma chamada para o procedimento searchpause.

```
procedure searchpause;
begin
  pauseFlag := true;
  while pauseFlag do application.ProcessMessages;
end;
```

A bandeira de pausa é apagada pressionando-se a barra de "espaço".

```
procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if searchbusy then
    if key = VK_SPACE then pauseflag := false;
end;
```

O sinalizador searchbusy é definido no início do GOsearch e é apagado após o término da pesquisa. Sua finalidade é bloquear eventos indesejados durante a pesquisa.

Para obter detalhes, consulte o código-fonte.



Lazarus **HANDBOOK (Pocket)**
+ SUBSCRIPTION 1 YEAR

- Lazarus **Handbook**
- Printed in black and white
- PDF Index for keywords
- Almost 1000 Pages
- Including 40 Examples
- **Blaise Pascal Magazine**
- English and German
- Free Lazarus PDF Kit Indexer
- 8 Issues per year
- minimal 60 pages
- Including example projects and code

SPECIAL OFFER € 85

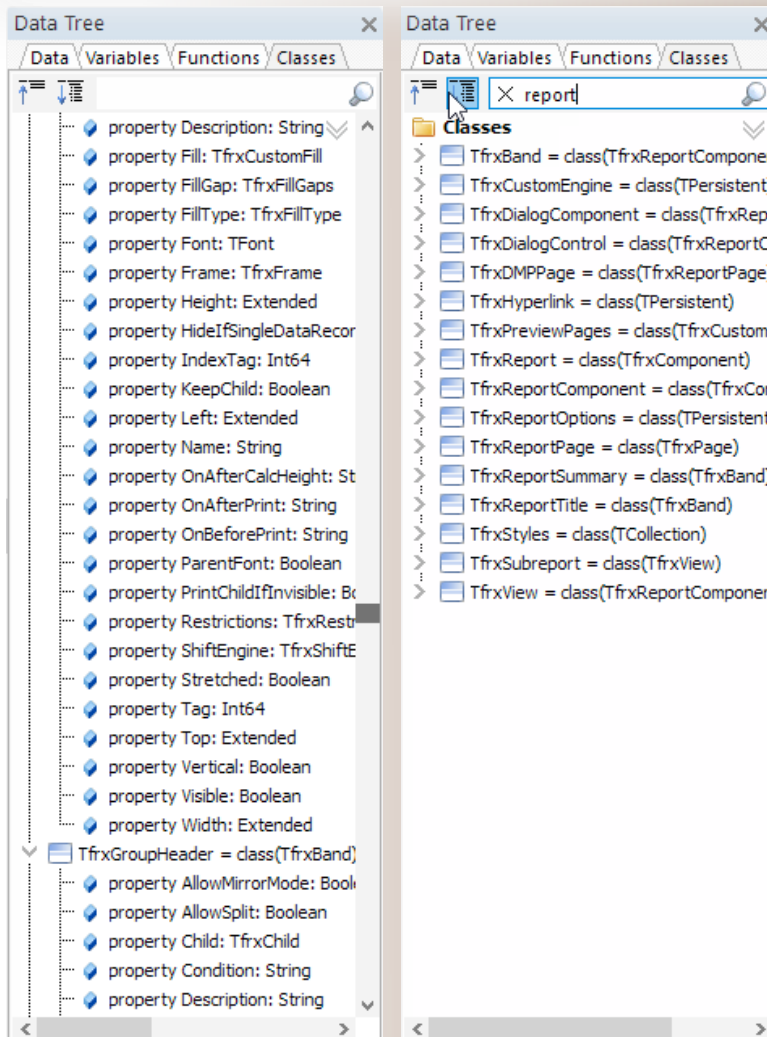


ATUALIZAÇÃO DOS PRODUTOS **Delphi** E **Lazarus** PARA A VERSÃO 2024.2

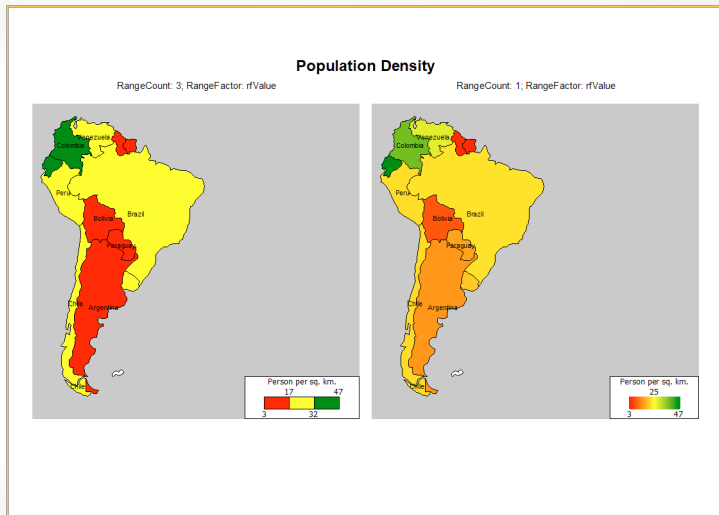
Essa atualização inclui melhorias e correções para toda a nossa linha de produtos **Delphi** e **Lazarus**. Um dos principais recursos dessa atualização é o novo pacote com componentes visuais **TfrTreeView**. O novo **TfrTreeView** permite que você crie rapidamente seus equivalentes personalizados do **TreeView**, expandindo a funcionalidade do seu aplicativo. Em comparação com o **TreeView** padrão, nosso componente tem navegação rápida e conveniente, recolhendo e expandindo a árvore de milhares de elementos. Isso aumenta a capacidade de resposta da interface do seu aplicativo para os usuários finais. E, é claro, o **TfrTreeView** é compatível com **VCL** e **Lazarus!**

FastReport VCL

Você pode experimentar a capacidade de resposta da interface no designer de relatórios do **FastReport VCL** ao trabalhar com uma grande quantidade de dados ou objetos, graças ao uso do novo componente **TfrTreeView**.

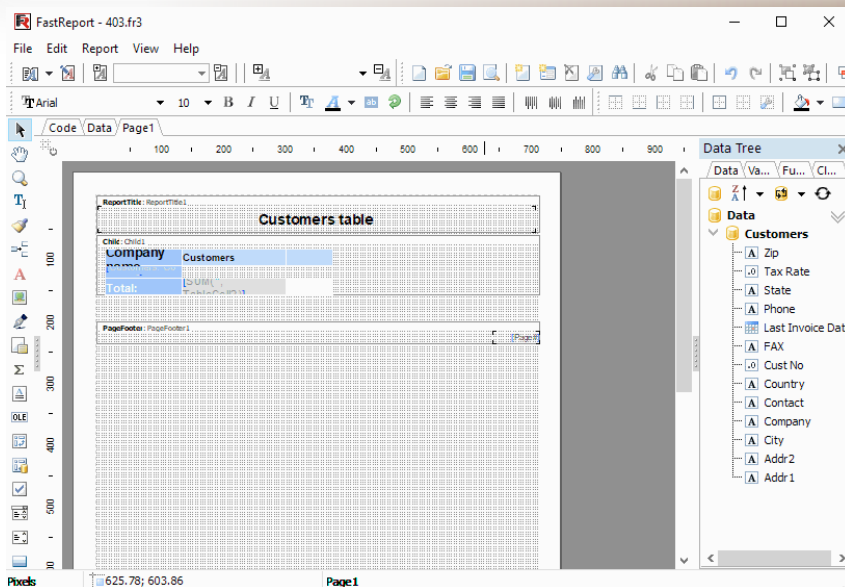


O suporte aos formatos GeoJSON e TopoJSON foi adicionado ao objeto de mapa.



Tratamento aprimorado de assinaturas digitais em PDF e outros arquivos. Nosso produto permite o máximo de flexibilidade na configuração de assinaturas digitais. Você tem acesso às seguintes configurações:

- Método de assinatura: FastReport, CryptoAPI, CryptoPRO
- Tipo de assinatura: CAdES-BES, CAdES-T, CAdES-X Type 1.
- Servidores de autoridade de registro de data e hora.
- Armazéns de certificados.
- Foi adicionado um modo de largura infinita para o objeto de tabela dinâmica. O tamanho da tabela aumenta de acordo com os dados que estão sendo gerados. Na janela de visualização, você pode ver uma página que se encaixa em todas as colunas da tabela. Enquanto isso, no modo normal, as colunas que não couberem na planilha serão movidas para a próxima página.



Os relatórios com páginas detalhadas retêm os estados das variáveis transmitidas para cada guia, permitindo a reconstrução individual de cada guia ao atualizar os parâmetros. Uma nova propriedade para códigos de barras lineares, `DigitsAutoFillMode`, foi adicionada. `dafmLeftOnly` preenche os dígitos ausentes do código de barras com zeros do lado esquerdo. `dafmRightChecksum` sempre adiciona uma soma de verificação zero quando há falta de dígitos exigidos pelo padrão de código de barras.



Nesta atualização, aprimoramos e resolvemos vários problemas com os filtros de exportação HTML5, SVG, PDF, XLSX e DOCX. As exportações de HTML5, SVG e PDF agora estão mais próximas do WYSIWYG completo. A qualidade das exportações de DOCX e XLSX foi aprimorada para as exportações de tabelas. Um caminho alternativo no Linux registrou melhorias gerais na operação do WINE no Linux.

FastQueryBuilder

O suporte ao **Lazarus** foi adicionado ao FastQueryBuilder - agora você pode se conectar e usá-lo em seus projetos **Lazarus**.

FastCube

Os erros encontrados foram corrigidos e a integração com a base de código comum foi aprimorada.

FastReport FMX

Foram adicionados dois novos filtros de exportação DOCX e XLSX - exportar relatórios para o formato que você precisa.



Starting with the release, we will **NO LONGER SUPPORT** development environments released before Embarcadero RAD Studio 10.4 for the **FireMonkey** platform.



BLAISE PASCAL MAGAZINE 118/119

Multi platform /Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js /
Databases / CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux



Lazarus HANDBOOK POCKET+PDF+ DOWNLOAD MAGAZINE SUBSCRIPTION

EX VAT AND SHIPPING PRICE: € 85,00



AI FPC



The screenshot displays the Lazarus IDE v3.99 interface. The top menu bar includes File, Edit, Search, View, Source, Project, Run, Package, Tools, Window, and Help. Below the menu is a toolbar with various icons for file operations and development tools. The main workspace is divided into three panes:

- Object Inspector:** Shows the component tree with 'Form1: TForm1' selected. The Properties panel below lists various attributes such as Align, AllowDropFiles, AlphaBlend, and Anchors.
- Source Editor:** Displays the source code for 'Unit1'. The code includes a unit declaration, an interface, a type definition for 'TForm1', and a procedure 'FormCreate'. A tooltip for 'Form1: TForm1' is visible, showing its position (256, 145) and size (320 x 240).
- AI Assistant conversation form:** A window titled 'AI Conversation' with a green background. It contains a chat interface with a system message: 'Hello there! How can I assist you today? If you have any questions or need help with something, feel free to ask. I'm here to help!'. Below this, there are two user prompts: 'If you're just looking to chat, we can talk about a variety of topics such as books, movies, music, or even science and technology. What would you like to discuss?' and 'Also, if you have any feedback or suggestions for me, I'd love to hear them! I'm constantly learning and improving to better serve you.' A 'say hello' button is located in the top right corner of the chat area.

A callout box points to the AI Assistant window with the text: 'Área de conversação para os AI Generator de sua escolha'. Another callout box points to the input field of the AI Assistant with the text: 'Digite aqui sua solicitação ou pedido'. The input field contains the text 'say hello' and a 'Prompt AI' button is visible to its right. A 'Configure...' button is located at the bottom right of the AI Assistant window.





```
AI Assistant conversation form
AI Conversation

Here is a simple example of a Lazarus Pascal code that displays a 'Hello, World!' message using the FMX (FireMonkey) framework. This example assumes you have the FMX unit installed.

''Lazarus
program HelloWorld;

uses
  {SAPPTYPE FMXMobile}

uses
  FMX.Types, FMX.Controls, FMX.Forms, FMX.GR;

var
  Form1: TForm1;

{$SR *.fmx}

procedure TForm1.FormCreate(Sender: TObject);
begin
  ShowMessage('Hello, World!', 'Hello World Exa
end;

end.

// Form1.fmx file
object Form1: TForm1
  Left = 0
  Top = 0
  Width = 320
  Height = 240
  Caption = 'Hello World Example'
  PixelsPerInch = 96
  OnCreate = FormCreate
  object Label1: TLabel
    ParentFont = False
    ParentStyleSettings = True
    Text = 'Hello, World!'
    Left = 56
    Top = 72
    Width = 100
    Height = 21
  end
end;

This code creates a simple form with a label disp
message box with the same message.

To run this code, save the code in a file named 'H
Then, open Lazarus, create a new project, and ad
project, you should see the "Hello, World!" mess

Here is a simple Lazarus Pascal code sample that
from the Lazarus VCL library.

''Lazarus
program HelloWorld;

uses
  Vcl.Forms, Vcl.Dialogs;

type
  TForm1 = class(TForm)
  procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

{$SR *.ifm}

procedure TForm1.FormCreate(Sender: TObject);
begin
  ShowMessage('Hello');
end;

begin
  Application.Initialize;
  Form1 := TForm1.Create(nil);
  Application.Run;
end.

Your prompt: |
```

Resultado de um prompt dado: mostra uma mensagem de olá

Você terá que instalar o servidor primeiro e depois iniciá-lo.

Settings cannot be modified while the server is running

Se você vir o botão vermelho Stop server (Parar servidor), o servidor está pronto para entrar em ação.

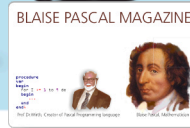
The screenshot shows the AI Assistant interface with two main panels: 'Server Options' and 'Server Logs'. The 'Server Options' panel includes fields for IP address (127.0.0.1), port (1337), and API Prefix (/v1). It also has checkboxes for 'Cross-Origin-Resource-Sharing (CORS)' and 'Verbose Server Logs'. The 'Start Server' button is blue, and the 'Stop Server' button is red. The 'Server Logs' panel shows a log of messages, including 'Starting JAN 27 13:23:07 API listening at 127.0.0.1:1337' and 'Request completed'. A warning message 'Settings cannot be modified while the server is running' is overlaid on the 'Stop Server' button.

EXECUTING PROGRAMS ON THE SERVER IN

By Michael Van Canneyt



ARTICLE PAGE 1 / 18



ABSTRACT

In this article we show how to give the user of a browser-based program feedback from long-running processes on the server, using 2 components: one in **PAS2JS**, one in Free Pascal/Lazarus.

1 INTRODUCTION

When using a web-based program, not everything can be done in the browser.

Often, tasks are executed through some (Post Request or Remote Procedure Call) mechanism on the webservice. This can be a simple task such as executing an SQL statement on a database and returning a result, or it can be a more complicated and time-consuming task such as making a backup of a database, indexing PDF files, compiling a source code project and running a test suite, or even installing a new operating system. The output of these remote programs should all be presented to the user.

To keep programs scalable, these tasks should be performed on the server. If a request is already a long time, so executing a time-consuming task and waiting for the result of a single HTTP request is not a good idea:

the HTTP server is occupied with this request, the browser may decide to time-out your request.

Much better is to start the process using a long request, and use a polling mechanism to poll the status of the executed process. In this article we present one such mechanism.

2 ARCHITECTURE

The solution we present here consists of 2 components. One component which is used on the server, and which can be used to start a process, capture its output and poll for the status of the process. The other component takes care of the polling process on the client.

These components are ignorant of the communication mechanism between browser and server, this means that they do not implement the actual RPC calls used to start the process: There are many possible mechanisms, and some may be more suitable for your purpose than others.

The components are called **TProcessCapture** for the server part and **TProcessCapturePoller** for the client (**PAS2JS**) part. The server part takes care of executing a program and redirecting the output to a file, the client part implements the polling mechanism and some callbacks to handle the actual server calls and the result. We'll demonstrate both components with a simple set of programs:

- A test program to be executed. It is used for demonstration purposes only.
- A HTTP server program that allows to serve HTML files and that offers an RPC mechanism to start the test program and handle status requests. A simple **PAS2JS** program that will run in the browser and which will remotely execute the test program. It will show the output of the test program in the browser.

We'll start with the test program.



With Highlight the result on search





PREVIEW AI FPC

Q Search or paste Hugging Face URL [Import Model](#)

All Models

Model Name	Size	Recommendation	Action
Gemma 2 2B Q4			Download
Llama 3.1 8B Instruct Q4	4.58GB	Recommended	Download
Gemma 2 9B Q4	5.36GB	Recommended	Download
Codestral 22B Q4	12.42GB	Recommended	Download
Gemma 2 27B Q4	15.46GB	Recommended	Download
Command-R v01 34B Q4	20.02GB	Slow on your device	Download

System Monitor Jan v0.5.4

Lista de servidores que podem ser instalados localmente. É necessário ter uma boa placa de vídeo

General

My Models

Appearance

Keyboard Shortcuts

Advanced Settings

Extensions

Model Providers

Anthropic

Cohere

Groq

Martian

MistralAI

TensorRT-LLM

NVIDIA NIM

OpenAI

OpenRouter

Triton-TRT-LLM

Core Extensions

Model Management

System Monitoring

Q Search or paste Hugging Face URL [Import Model](#)

Llama.cpp (Nitro)

Mistral 7B Instruct Q4 mistral-ins-7b-q4 4.07GB Inactive

Anthropic

Cohere

Groq

Martian

Nvidia

Mistral

OpenAI

OpenRouter

<https://jan.ai/> é o endereço da Web desse serviço

System Monitor Jan v0.5.4

Este é o serviço de IA que instalei



EUGENE KRYUKOV †

LAMENTAMOS PROFUNDAMENTE INFORMAR QUE FOMOS INFORMADOS SOBRE O RECENTE FALECIMENTO REPENTINO DO EXTRAORDINARIAMENTE TALENTOSO EUGENE KRYUKOV.

Eugene era muito conhecido e respeitado na comunidade **Delphi** em todos os níveis. Muitas vezes, ser solicitado a enumerar as conquistas de um desenvolvedor em uma vida tão curta de codificação pode ser simples, mas Eugene foi tão prolífico e operou em um nível tão alto e consistente de inovação, criatividade e conhecimento técnico que é quase impossível fazer justiça a uma vida tão completa e extraordinariamente talentosa.



Eugene criou originalmente um produto chamado VGSCENE, que foi adquirido pela Embarcadero. O VGSCENE foi renomeado e expandido como **FireMonkey** sob a direção de Eugene. Ele continuou a trabalhar no **FireMonkey**, incluindo o **FireMonkey 3D** e, mais tarde, foi responsável por um sucesso após o outro.

Como disse David Millington, "Eugene era um desenvolvedor muito talentoso. Ele tinha uma visão real da modernidade dos sistemas de computação e trouxe isso para o **Delphi** por meio de um forte foco em plataformas cruzadas e várias soluções diferentes para isso - algumas pessoas teriam ficado satisfeitas com uma, o que por si só é uma conquista para a vida toda, mas por meio do VGScene/**FMX**, do Turbo Cocoa e do CrossVCL, ele construiu três".

EUGENE, UMA PERSONIFICAÇÃO DA PALAVRA "DESENVOLVEDOR"

Para a maioria das pessoas, isso seria suficiente para relaxar e dizer: "Fiz algo incrível".

Mas Eugene não era esse tipo de pessoa.

Ele trabalhou ativamente no FL Studio, um aplicativo de criação de música com um design maravilhoso e extremamente popular que envergonha seus concorrentes.

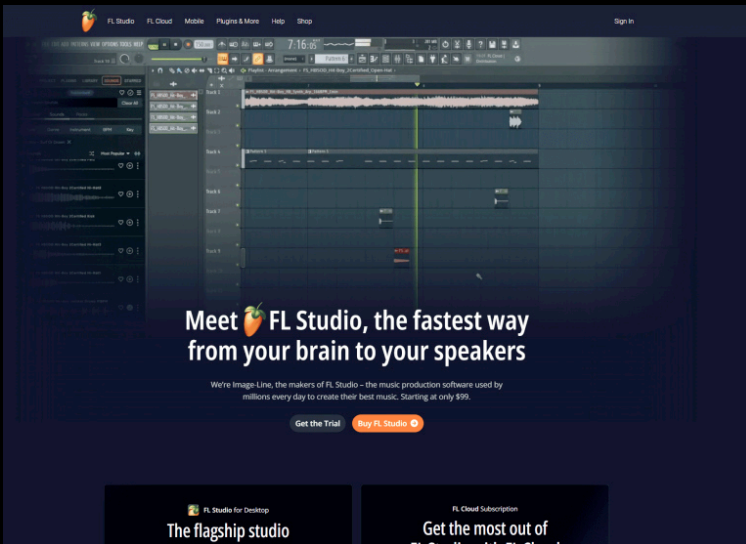


Eugene Kryukov à direita com seu bom amigo Alexey Shargin em uma foto da equipe da Embarcadero

No entanto, apesar dessa capacidade técnica abrangente e da enorme produção de criatividade, Eugene permaneceu firme e era popular, conseguindo conhecer e fazer amizade com programadores de todos os níveis, dentro e fora da Embarcadero. Eugene promoveu atitudes de desenvolvedor e criou uma tecnologia básica fundamental que tem sido usada diretamente e da qual muitos milhares de pessoas dependem e que permanecerá assim em um futuro próximo. Ele teve um impacto enorme. Sua influência permanecerá visível e os resultados de seu trabalho serão usados nas próximas décadas.

TODOS NÓS TEMOS QUE PARTIR, EVENTUALMENTE, MAS É SEMPRE CEDO DEMAIS

Todos nós somos extremamente gratos a Eugene e a tudo o que ele foi capaz de realizar em uma vida tão curta, que terminou tão repentinamente.



O TRABALHO DE EUGENE KRYUKOV TORNOU O SEU POSSÍVEL

Se você já usou o **FireMonkey FMX**, seu código foi tocado pelo talento de Eugene. Se você já teve um aplicativo **Delphi** GUI funcionando em uma máquina Linux, então os componentes **FMXLinux** dele tornaram isso possível. CrossVCL, a solução surpreendentemente inteligente para obter aplicativos VCL (uma solução somente para Windows)



Yuriy Kotsarenko, Eugene Kryukov e Dmitry Arefiev no Delphi Day em Piacenza, Itália

COMPILAÇÃO DE UM APLICATIVO ^{FMX} PARA O NAVEGADOR

POR MICHAEL
VAN CANNEYT



THE RED FIRE MONKEY

Design gráfico:
Anna Simanovych
Direção de arte:
Oleksii Simanovych



COMPILAÇÃO DE UM APLICATIVO PARA O NAVEGADOR

ARTIGO
PÁGINA 2 / 26

Por Michael van Canneyt



SINOPSE

O **FireMonkey (FMX)** é a estrutura de GUI multiplataforma do **Delphi**. Ele permite que você compile seu aplicativo para todas as principais plataformas nativas: Windows, MacOS, Android e iOS. E agora, graças ao **Free Pascal**, também para o navegador.

[Se possível, é melhor visualizar este artigo como páginas duplas no PDF](#)

1 INTRODUÇÃO

Há muito tempo, o **Delphi** usa o **FireMonkey (FMX)** para oferecer uma **GUI** para suas plataformas compatíveis que não sejam o **Windows**.

No entanto, uma plataforma importante está faltando: o navegador.

O criador do **FireMonkey**, **Eugene Kryukov**, estava ciente de que a equipe do **Free Pascal** - patrocinada pelo **Tixeo.com** - estava empreendendo um esforço para portar o **FireMonkey** para o navegador. Sua morte prematura, apenas uma semana antes da primeira execução bem-sucedida da demonstração **FMX** de "todos os controles" no navegador, significa que ele não pôde ver sua criação sendo executada no navegador.

Depois de mais de um ano de trabalho, agora é possível compilar um aplicativo **FMX** para **WebAssembly** usando o **Free Pascal** e executá-lo no navegador.

Para isso, muitas coisas tiveram que ser feitas no **Free Pascal**:

- 1 O **RTTI** estendido teve de ser integrado à ramificação principal.
- 2 O compilador teve que ser estendido para habilitar todos os recursos do **Object Pascal** para **WebAssembly** (*notavelmente, threads e Invoke ()*).
- 3 A **RTL** distribuída com o **FPC** teve que se tornar mais compatível com a **RTL** do **Delphi**: namespaces foram introduzidos e muitas unidades ausentes foram adicionadas.
- 4 Foram necessários alguns patches para adaptar o **FMX** para que ele pudesse ser compilado pelo **FPC**.
- 5 Para facilitar o trabalho no navegador, foi adicionado o suporte a **DWARF** para o navegador (*atualmente, apenas o Chromium permite depurar seu código pascal no navegador*).
- 6 Habilitar o suporte ao sistema de arquivos para **WebAssembly** em execução no navegador.
- 7 Dar acesso ao **WebAssembly** ao **DOM** do navegador e, em particular, à tela **HTML**.

Depois de todo esse trabalho, agora você pode compilar um aplicativo **FMX** básico e executá-lo no navegador. *Veja a figura 1 na página 3*. É claro que muitas coisas ainda não são possíveis. Algumas devido à natureza do navegador, outras devido à falta de recursos no **FPC**. O que (*ainda*) não é possível?



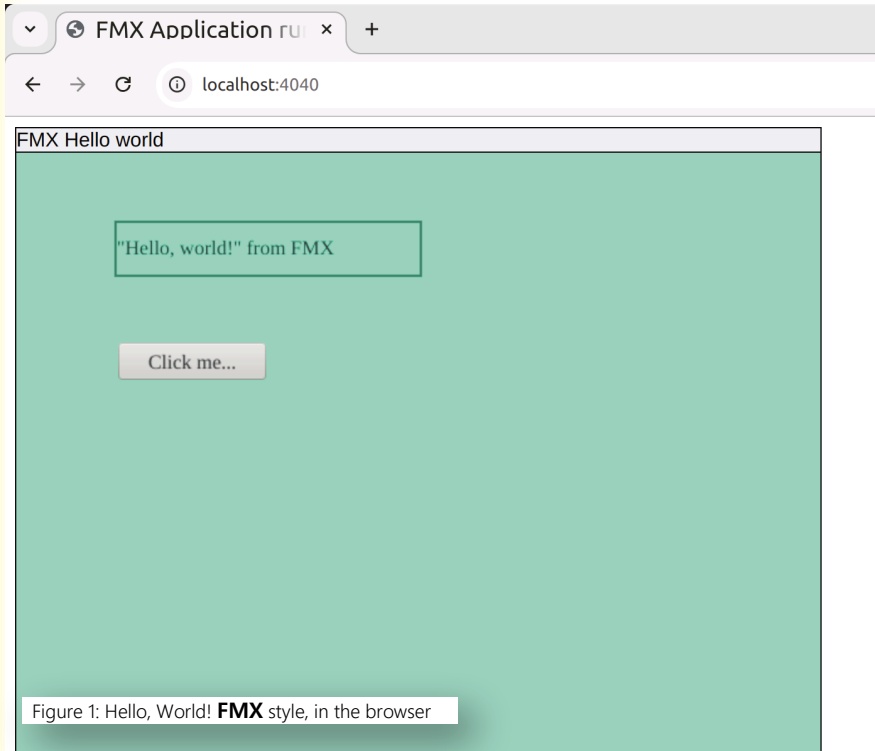


COMPILAÇÃO DE UM APLICATIVO PARA O NAVEGADOR

ARTIGO
PÁGINA 3 / 26

❶ INTRODUCTION (CONTINUATION)

WHAT IS NOT (YET) POSSIBLE ?



- ❶ **Acesso ao banco de dados.** O `TDataSet` está disponível, mas não existe (*ainda*) uma camada de dados assíncrona. É possível carregar dados de um arquivo, se você tiver um `TDataSet` capaz de fazer isso.
- ❷ **Ligações ao vivo** - O **Free Pascal** ainda não possui os objetos básicos usados nas ligações ao vivo.
- ❸ **Diálogos do sistema**, como "abrir arquivo", "salvar arquivo", "imprimir" etc. Isso ocorre porque o navegador não expõe as **APIs** subjacentes necessárias e as APIs disponíveis são assíncronas, o que as torna incompatíveis com as **APIs** básicas definidas pela estrutura **FMX**.
- ❹ **Renderização 3D.** Isso está no roteiro, o navegador oferece WebGL e, portanto, teoricamente é possível adicionar isso.

Se essas limitações forem aceitáveis (*algumas podem ter sido removidas quando você ler este artigo*) e você tiver uma licença **Delphi** que permita o acesso aos códigos-fonte do **FMX**, poderá compilar um programa **FMX** para **WebAssembly**.

Como compilar seu próprio aplicativo **FMX** para **WebAssembly**?

Para poder compilar um aplicativo **Delphi** para **FMX**, são necessárias várias etapas.





② CRIAR UM COMPILADOR CRUZADO DE WEBASSEMBLY.

A primeira coisa a fazer é criar um ambiente de desenvolvimento onde você possa compilar para o : Você precisa de um compilador cruzado de **WebAssembly**. Se você já tiver criado um, poderá pular esta etapa ou repeti-la para garantir que tenha os recursos mais recentes.

As instruções para criar um compilador cruzado podem ser encontradas em:

<https://wiki.freepascal.org/WebAssembly/Compiler>

Elas também foram publicadas em um artigo anterior.

Para começar, você precisa ter a versão lançada do **FPC** instalada em seu sistema.

A criação de um compilador cruzado para **WebAssembly** começa com o download da versão de desenvolvimento do compilador usando o git. Portanto, você precisa de um cliente git, qualquer cliente git padrão deve ser adequado.

Em um diretório de sua escolha, execute o seguinte comando:

```
git clone https://gitlab.com/freepascal.org/fpc/source.git fpc
```

Em seguida, você precisa se certificar de que os binários instalados pelo **FPC** estejam em seu **PATH**.

Quando estiver tudo certo, vá para o diretório de nível superior dos códigos-fonte do **FPC** que você clonou com o git e execute o seguinte comando. A barra invertida significa que o comando é continuado na linha seguinte; você pode omiti-la se digitar as duas linhas exibidas em uma única linha:

```
make clean crossall OPT='-dSKIP_INTERNAL20231102' \ PP=ppcx64 OS_TARGET=wasi CPU_TARGET=wasm32
```

Isso recompilará o compilador e todos os pacotes.

O `-dSKIP_INTERNAL20231102` é uma opção para desativar um erro específico do compilador, que pode ocorrer durante a compilação do **FMX**.

Espera-se que ele desapareça mais tarde.

Se a compilação for bem-sucedida, você poderá instalar essa nova versão com o seguinte comando:

```
make crossinstall PP=ppcx64 OS_TARGET=wasi CPU_TARGET=wasm32
```

Com isso, você terá um compilador cruzado em funcionamento.

③ CRIE UMA RTL COMPATÍVEL COM O DELPHI.

No entanto, ter um compilador cruzado não é suficiente.

Com o compilador cruzado, você pode compilar códigos-fonte do **FPC**, e o resultado funcionará no navegador.

Mas para compilar o **FMX** e um programa **FMX**, precisamos ter também uma **RTL** compatível com o **Delphi**.

Os códigos-fonte da **RTL** e dos pacotes do Free Pascal podem ser compilados em um dos dois modos:

- O **RTL** compatível com versões anteriores do **FPC**: uma string é uma string de 1 byte. Não são usados namespaces para as unidades.
- O **RTL** compatível com **Delphi**: uma string é uma string de 2 bytes. Os namespaces são usados.

Para criar a **RTL** compatível com o **Delphi**, as instruções podem ser encontradas em:

https://wiki.freepascal.org/FPC_Unicode_RTL

A primeira coisa a fazer é criar três arquivos, necessários para o suporte a sub-targets. O suporte a sub-targets permite compilar e instalar uma **RTL** com configurações diferentes e usá-las em suas compilações sem muito esforço.

Ele foi introduzido para facilitar a compilação e a instalação da **RTL** e dos pacotes.

Você pode pensar nisso como uma configuração de compilação como a oferecida no **Delphi** ou no **Lazarus**, mas no nível do compilador.





❶ CREATE A Delphi-COMPATIBLE RTL.

Precisamos de 3 sub-targets e, para isso, precisamos criar 3 configurações nomeadas. Para sistemas operacionais do tipo Unix (*incluindo o Mac*), esses três arquivos estão localizados em seu diretório pessoal e são chamados de:

```
.fpc-unicodertl.cfg  
.fpc-browser.cfg  
.fpc-unicodertl-browser.cfg
```

Para sistemas operacionais do tipo Windows, os arquivos estão localizados ao lado do binário do compilador (**ppcrosswasm32**) e são chamados de

```
fpc-unicodertl.cfg  
fpc-browser.cfg  
fpc-unicodertl-browser.cfg
```

O primeiro desses arquivos (`.fpc-unicodertl.cfg`) contém as duas linhas a seguir:

```
-dUNICODERTL  
-Municodestrings
```

Isso é necessário para a criação de uma **RTL** compatível com o **Delphi**: cadeias de caracteres de 2 bytes e arquivos de unidade namespaced.

O segundo desses arquivos (`.fpc-browser.cfg`) contém a seguinte linha:

```
-CTwasmexceptions
```

Isso permite o uso do suporte nativo a exceções do **WebAssembly**.

Com o tempo, essa opção desaparecerá, já que todos os mecanismos de **WebAssembly** atuais oferecem suporte a exceções de **WebAssembly**.

O terceiro arquivo (`.fpc-unicodertl-browser.cfg`) combina os dois arquivos anteriores:

```
#INCLUDE /home/michael/.fpc-unicodertl.cfg  
#INCLUDE /home/michael/.fpc-browser.cfg
```

Obviamente, você deve alterar o caminho `/home/michael/` para o caminho correto para sua configuração. Depois que esses arquivos forem criados, a **RTL** e os pacotes compatíveis com o **Delphi** poderão ser estruturados. No diretório de nível superior da árvore de código-fonte do FPC, digite os seguintes comandos:

```
cd RTL  
make clean all SUB_TARGET=unicodertl-browser FPC_DOTTEDUNITS=1 \ OPT="-g"  
PP=ppcrosswasm32  
make install SUB_TARGET=unicodertl-browser FPC_DOTTEDUNITS=1 \  
PP=ppcrosswasm32  
cd ..
```

Novamente, a barra invertida no final da linha significa que o comando continua na próxima linha. Se tudo ocorreu sem erros, então as unidades básicas da **RTL** foram compiladas com namespaces e instaladas em um diretório dedicado (*separado do diretório padrão*).

O restante das unidades (*e há muitas*) pode ser compilado e instalado de forma semelhante

```
cd packages  
make clean all SUB_TARGET=unicodertl-browser FPC_DOTTEDUNITS=1 \ OPT="-g"  
PP=ppcrosswasm32  
make install SUB_TARGET=unicodertl-browser FPC_DOTTEDUNITS=1 \  
PP=ppcrosswasm32  
cd ..
```

Observe que especificamos `-OPT=-g`, o que significa que a opção `-g` é adicionada à linha de comando do compilador e instrui o compilador a gerar informações de debug.





④ PREPARANDO AS FONTES FMX

Para compilar um programa **FMX**, você precisa, é claro, dos códigos-fonte do **FMX**.

O **FMX** é um produto protegido por direitos autorais, com os direitos autorais pertencentes à **Embarcadero**. Isso significa que, obviamente, a equipe do **Free Pascal** não pode distribuir o **FMX**. Você só pode prosseguir se tiver uma licença **Delphi** válida que permita o uso do **FMX**.

Infelizmente, os códigos-fonte do **FMX** não são compilados com o **Free Pascal** imediatamente.

Os códigos-fonte do **FMX** usam algumas construções **Pascal** que não são suportadas pelo **Free Pascal**:

- uma classe não genérica e uma classe genérica não podem ter o mesmo nome no **Free Pascal**.
O **Delphi** permite isso, e é usado em seu sistema de mensagens.
- Variáveis inline.

O uso desses recursos não é muito difundido e o número de alterações não é muito grande, portanto, é possível criar um patch (*um arquivo diff*).

Esse patch foi criado. O patch é pequeno o suficiente para que seja completamente impossível reconstruir os códigos-fonte completos do **FMX** a partir dele.

Entramos em contato com a **Embarcadero** para perguntar se seria considerado uma violação de direitos autorais se distribuíssemos um patch para seus códigos-fonte.

Felizmente, a resposta foi negativa, portanto, podemos disponibilizar o patch.

Para usar esse patch, a melhor abordagem é fazer uma cópia dos códigos-fonte do **FMX** e aplicar o patch em:

<https://gitlab.com/freepascal.org/fmx-using-fpc>

Esse repositório contém todos os arquivos necessários para compilar um aplicativo **FMX** para o **WebAssembly**. O patch pode ser aplicado à versão do **FMX** fornecida com a versão 12.1 do **Delphi**. Ele pode ser bem-sucedido em outras versões.

Por exemplo, ele funciona principalmente na 11.3, apenas uma pequena parte precisa ser verificada manualmente.

Para aplicar a correção, baixe o arquivo de correção e coloque-o no diretório em que você copiou os códigos-fonte do **FMX**.

Em seguida, na linha de comando, vá até a cópia dos códigos-fonte do **FMX** e execute o seguinte comando:

```
patch < fmx-fpc-1210.diff
```

O **Free Pascal** no Windows distribui uma versão da ferramenta de correção, que pode ser usada para executar o comando acima.

Se tudo tiver ocorrido corretamente, você deverá ter uma versão do **FMX** que compila com o **FPC**.

No entanto, devido à estrutura do **FMX**, não podemos testar isso: se você tentar compilá-lo neste momento, a compilação falhará com uma mensagem de unidade ausente.





5 A TELA DO WEBASSEMBLY

O **FireMonkey** fornece uma abstração das APIs da GUI do sistema operacional. Basicamente, ele define algumas interfaces abstratas que devem ser implementadas para cada plataforma. As mais importantes são a Canvas API e a Window API. A Window API é responsável pela criação de uma janela e pelo tratamento das mensagens que o sistema envia a essa janela: movimentos do mouse, cliques do mouse, pressionamentos de tecla e assim por diante.

A **Window API** e a **Canvas API** foram criadas para uso em **WebAssembly**.

Infelizmente, essa API deve ser adicionada nos próprios códigos-fonte do **FMX**.

Portanto, o patch acima incluirá uma referência a alguns arquivos que ainda não discutimos e não será compilado até que a próxima etapa seja concluída: adicionar as APIs Canvas e Window.

As **APIs Window e Canvas** para **WebAssembly** foram criadas com base nas **APIs Window e Canvas** que foram desenvolvidas para o Projeto **Fresnel**, o projeto que fornece aos desenvolvedores do **FPC** e do

Lazarus uma estrutura de interface do usuário baseada no uso de **CSS**.

Um aplicativo criado com o **Fresnel** é executado no navegador usando essa **API** e, como seria bobagem recriar o que já existe, a tela do **Fresnel** é reutilizada para a tela do **FMX**.

Essa **API** contém duas partes:

- A **API** de backend do **Fresnel WebAssembly**: as definições da **API** do **WebAssembly** e a implementação dessa definição no lado do navegador do **Pas2JS**.
- A **API** de **Canvas e Window** do **FMX** que se baseia no backend do **Fresnel**.

No repositório, o lado do **WebAssembly** de ambas as partes está no diretório `Src/Wasm`, nos subdiretórios **Fresnel** e **FMX**, respectivamente.

A parte de hospedagem do navegador está em `Src/Pas2JS` e é necessária para implementar de fato a **API** do navegador.

A parte do **Fresnel** é uma cópia do que pode ser encontrado no repositório do **Fresnel** e está incluída por conveniência.

Portanto, para prosseguir, você precisa verificar esse repositório:

```
git clone https://gitlab.com/freepascal.org/fmx-using-fpc.git
```

E adicione os diretórios necessários à sua configuração do **FPC**.

Se estiver usando o **Lazarus**, é fornecido um pacote que faz tudo para você: `fpcforfmx.lpk`, localizado no diretório `Src/Wasm`.

Você pode adicioná-lo como uma dependência a um projeto do **Lazarus** para compilar seu projeto **FMX**.

O diretório `Src/Wasm/Package` contém um pacote **Lazarus** (`fmx.lpk`)

no qual estão incluídos todos os arquivos **FMX** que são conhecidos por compilar.

Esse pacote também pode ser adicionado ao seu projeto **Lazarus FMX**, de modo que o compiler encontrará todos os arquivos necessários para compilar seu projeto.





⑥ HOSPEDAGEM DE UM PROGRAMA FMX NO NAVEGADOR (CONTINUAÇÃO)

Um programa de **WebAssembly** (*comumente chamado de "módulo"*) é um bytecode que é carregado em um programa de hospedagem, que então executa as instruções do bytecode.

Por padrão, o bytecode do **WebAssembly** não tem acesso ao ambiente no qual o programa de hospedagem está sendo executado.

O programa de hospedagem pode (*e deve*) fornecer APIs ao módulo do **WebAssembly** para interagir com o ambiente. O módulo do **WebAssembly** tem uma lista de chamadas de **API** que ele espera receber e, se uma dessas chamadas de **API** estiver faltando, ele não poderá iniciar: semelhante a ter dependências em bibliotecas carregáveis dinamicamente em um ambiente nativo.

Um programa típico requer acesso ao sistema de arquivos, acesso ao relógio e, no caso de um aplicativo **FMX**, uma tela e alguns eventos da interface do usuário. O acesso ao sistema de arquivos é definido no padrão **WASI**, e é fornecido pelo compilador **PAS2JS do Free Pascal**. O **PAS2JS** também fornece acesso direto ao tráfego HTTP e WebSocket, bem como ao mecanismo **RegExp** do navegador. Além disso, por meio do **JOB - Javascript Object Bindings** - você tem acesso a todas as **APIs** do navegador.

As unidades **PAS2JS** que disponibilizam uma tela para um **WebAssembly** foram desenvolvidas para o projeto Fresnel e são reutilizadas para permitir que o **FMX** acesse a tela do navegador. Há 2 unidades:

fresnel.wasm.shared algumas constantes e tipos básicos usados na API.

fresnel.pas2js.wasmapi

A implementação real da **API**. Aqui você encontrará chamadas para um timer, alocação de uma tela e as operações usuais de desenho na tela.

Em essência, ela permite que o módulo **WebAssembly** faça tudo o que pode ser feito com uma tela **HTML**, o que é bastante.

Essas duas unidades estão disponíveis no repositório mencionado acima, na pasta `Src/Pas2JS`.

O **PAS2JS** oferece um ambiente completo para carregar módulos de **WebAssembly** e disponibilizar **APIs** para os módulos carregados: as classes `TWasiHost` e `TWasiEnvironment`.

A **API** foi discutida em outros artigos.

Usando essa API, é possível criar um programa de hospedagem genérico para um módulo de **WebAssembly** que executa um programa **FMX**.

Esse programa de hospedagem foi adicionado ao repositório acima.

É um programa **PAS2JS** padrão, que usa a classe de aplicativo desenvolvida para hospedar programas do **WebAssembly**.

Explicaremos a estrutura desse programa para que você possa adaptá-lo às suas necessidades.

Como de costume, temos de definir uma classe de aplicativo e,

como queremos ter suporte para carregar módulos do , ela deve ser um descendente de

`TBrowserWasiHostApplication`

```
TFMXHostApplication = class(TBrowserWasiHostApplication)
Private
    FZenFS      : TWasiZenFS;
    FFresnelApi : TWasmFresnelApi;
    FRegexp     : TWasmRegExpAPI;
Public
    constructor Create(aOwner : TComponent); override;
    procedure RunWasm; async;
    procedure DoRun; override;
end;
```





COMPILAÇÃO DE UM APLICATIVO PARA O NAVEGADOR

ARTIGO
PÁGINA 9 / 26

Na definição, você pode ver 3 campos: cada campo conterá uma classe que fornece uma funcionalidade específica ao programa:

O **TWASIZenFS** cria um sistema de arquivos virtual no navegador, usando o **ZenFS**. Esse sistema de arquivos poderá ser acessado pelo programa **WebAssembly**.

TWasmFresnelApi

A API Fresnel, que é o que o **FMX** precisa para gerar uma interface do usuário.

TWasmRegExpAPI

Acesso ao mecanismo de expressão regular do navegador.

O **FMX** usa expressões regulares, por exemplo, no componente de edição, para poder avaliar a entrada.

Essas classes são instanciadas e configuradas - onde mais - no construtor da classe do aplicativo, que começa criando o suporte ao sistema de arquivos e registrando-o no ambiente **WASI** nas primeiras linhas do código do construtor:

```
constructor TFMXHostApplication.Create (aOwner : TComponent) ;
begin
  inherited Create (aOwner) ;
  // Create and register the filesystem
  FZenFS := TWASIZenFS.Create ;
  WasiEnvironment.FS := FZenFS ;
  // The fresnel API.
  FFresnelApi := TWasmFresnelApi.Create (WasiEnvironment) ;
  FFresnelApi.CanvasParent := GetHTMLElement (' desktop ' ) ;
  FFresnelApi.CreateDefaultCanvas := True ;
  FFresnelApi.MenuSupport := True ;
  // The regular expression engine.
  FRegex := TWasmRegExpAPI.Create (WasiEnvironment) ;
  // An FMX program is a library.
  RunEntryFunction := '_initialize' ;
  if Assigned (hostConfig) then
  begin
    WasiEnvironment.LogAPI := HostConfig.logWasiAPI ;
    FFresnelApi.LogAPICalls := HostConfig.logFresnelAPI ;
    WasiEnvironment.Environment.Add (' FMX_LOGLEVEL=' + HostConfig.FMXLogLevel ) ;
  end ;
end ;
```





⑥ HOSPEDAGEM DE UM PROGRAMA FMX NO NAVEGADOR (CONTINUAÇÃO)

Da mesma forma, a **API Fresnel** é criada e configurada: a **API Fresnel** precisa de uma tag HTML sob a qual criará e posicionará as janelas que o **FMX** cria. Habilitamos o suporte a menus: o menu de um aplicativo **FMX** é renderizado em HTML. Dessa forma, é possível oferecer suporte a ambientes de menu como o **Mac** ou sistemas operacionais móveis.

A última **API** necessária é o mecanismo de expressão regular. Quando o suporte à expressão regular é criado, a `RunEntryFunction` é definida. Quando o módulo **WebAssembly** é iniciado, a função `RunEntryFunction` é a função que é chamada primeiro. Um programa **FMX** é, na verdade, uma biblioteca, e a biblioteca é inicializada com a função `inicialize`.

Por fim, o nível de registro é definido para as várias **APIs**, o que é útil para depuração. As configurações são lidas de um objeto **JSON** definido em um arquivo `hostconfig.js` que está incluído no arquivo HTML. Essa abordagem permite configurar o aplicativo host e definir qual **WebAssembly** precisa ser carregado sem a necessidade de recompilar o aplicativo host.

Como de costume, o método `DoRun` do aplicativo deve ser sobrescrito para implementar a funcionalidade do aplicativo.

```
procedure TFMXHostApplication.DoRun;
```

```
begin
  RunWasm;
end;
```

A função `RunWasm` é declarada como assíncrona, informando ao navegador que ele não precisa esperar pelo retorno. Isso nos permite configurar o sistema de arquivos **zenfs**: o sistema de arquivos **zenfs** é inicializado com o armazenamento da Web como backend, o que significa que os arquivos são armazenados no armazenamento local do navegador:

```
procedure TFMXHostApplication.RunWasm;

var
  wasm : String;
begin
  Terminate;
  await (TJSObject, ZenFS.configure (
    New ([
      'mounts', New ([
        '/' , DomBackends.WebStorage
      ])
    ]))) ;
  if Assigned (HostConfig) and isString (HostConfig.wasmFilename) then
    Wasm:=HostConfig.wasmFilename
  else
    begin // Allow to load file specified in hash: index.html#mywasmfile.wasm
      Wasm:=ParamStr (1);
      if Wasm='' then
        Wasm:=' HelloWorldFMX.wasm' ;
    end;
    StartWebAssembly (Wasm, true);
  end;
```





6 HOSTING A FMX PROGRAM IN THE BROWSER (CONTINUATION)

Depois disso, o objeto de configuração e o URL são examinados para ver qual arquivo deve ser carregado. Você pode especificar o nome do arquivo no objeto de configuração ou anexá-lo como hash no URL (**#wasmfilename**)

Por fim, o aplicativo é criado e iniciado:

```
var
  Application : TFMXHostApplication;
begin
  Application := TFMXHostApplication.Create(nil);
  Application.Initialize;
  Application.Run;
end.
```

É isso aí. Com o programa acima, você pode carregar qualquer programa **FMX** e executá-lo sem precisar recompilar o programa host. Ao adicionar algumas classes de **API** (*http, websocket e outras a seguir*), você pode fornecer mais funcionalidades ao programa **WebAssembly**.

Com todo esse trabalho, agora estamos prontos para compilar o primeiro aplicativo **FMX** para .

7 A PRIMEIRA TENTATIVA: 'HELLO, WORLD'

Para verificar se todo esse trabalho valeu a pena, criaremos um pequeno programa "hello world!".

Um formulário simples com um rótulo e um botão:

quando o botão é clicado, a cor de fundo do formulário é alterada.

Tendo em mente que é preciso tentar andar antes de correr, não usaremos um formulário, mas criaremos o formulário e os elementos nele contidos no código.

Antes de começar, é necessário mencionar um detalhe importante:

Qualquer programa de **WebAssembly** que precise executar um tipo de loop de mensagens precisa ser criado como uma biblioteca.

Isso ocorre porque o **WebAssembly** é arquitetonicamente mais uma biblioteca (*o termo usado é módulo*) do que um programa. É uma biblioteca que exporta funções. O ambiente do host (no nosso caso, o navegador) chama uma função e suspende todas as outras atividades até que a função do **WebAssembly** retorne.

Isso significaria que, no caso de um programa, todo o programa é executado antes que o navegador retome o processamento normal:

renderização do HTML, reação aos eventos do usuário etc.

Para o usuário, isso pareceria como se o navegador congelasse até que o programa saísse.

Poderíamos argumentar que o programa pode instalar alguns hooks de interface do usuário e depois sair do programa, mas isso não funcionará: quando o programa sair, o runtime do **Pascal** finalizará todas as unidades, inclusive a unidade do sistema:

quando um dos hooks instalados for chamado, o programa estará em um estado inutilizável.

Em vez disso, usamos um projeto de biblioteca:

Como mencionado anteriormente, a biblioteca expõe uma rotina 'initialize', que executa simplesmente o código de inicialização de todas as unidades e o bloco `begin..end` da biblioteca.

Portanto, devemos instalar os hooks necessários e criar o primeiro formulário no bloco `begin end`, depois disso, a função `initialize` sai e os hooks (*cliques do mouse etc.*) farão seu trabalho: como a biblioteca não foi finalizada, tudo continuará funcionando.





🔗 A PRIMEIRA TENTATIVA: 'HELLO, WORLD' (CONTINUAÇÃO)

Com esse conhecimento, podemos criar nosso primeiro aplicativo FMX:

uses

```
System.FPWideString  
, System.Unicode.Unicodeducet  
, System.CodePages.unicodedata  
, System.MonitorSupport  
, FpImage.Reader.PNG  
, FpImage.Reader.JPEG  
, FpImage.Reader.Bitmap  
, FMX.Wasm.WindowHandle  
, FMX.Controls.Wasm  
, FMX.Canvas.Wasm  
, FMX.Platform.Wasm  
, FMX.Forms  
, form.main in 'form.main.pas' {Form1};
```

```
{$R *.res}
```

begin

```
Application.Title := 'FMX Application running in the browser';  
Application.Initialize;  
Form1 := TForm1.CreateNew(Application, 0);  
Form1.Name := 'Form1';  
Application.MainForm := Form1;  
Form1.Show;  
Application.Run;
```

end.

O bloco begin..end é quase igual a um projeto normal de aplicativo de interface do usuário.

A única diferença é que, como não estamos usando um arquivo de formulário, precisamos criar o formulário manualmente e usar o construtor CreateNew: esse construtor cria o formulário, mas ignora o streaming do arquivo **fmx** associado ao formulário.

Ao usar a função Application.CreateForm, o fluxo não pode ser ignorado.

A cláusula uses merece uma atenção especial. O **Free Pascal** funciona de forma diferente do **Delphi** em muitos aspectos; um deles é o tratamento do unicode: A **RTL** do **Free Pascal** delega essa funcionalidade a plugins (os *chamados gerenciadores*), da mesma forma que o suporte a monitores:

o suporte ao registro TMonitor da unidade de sistema é implementado em outro lugar.

As unidades que implementam esses plug-ins devem ser inicializadas antes que o restante das unidades do programa seja inicializado, portanto, devem estar primeiro na cláusula uses do programa principal, e não em algum lugar oculto em outras unidades.

É por isso que as primeiras 4 unidades estão presentes na cláusula uses:

- O **System.FPWideString**, **System.Unicode.Unicodeducet** e o **System.CodePages.unicodedata** implementam o suporte a Unicode usando código em object pascal 100% nativo.
- **System.MonitorSupport** implementa o suporte a TMonitor.

O **FMX** depende dos serviços de back-end para oferecer suporte a imagens. Para o **WebAssembly**, a estrutura FpImage é usada para oferecer suporte a imagens. Novamente, essa é uma estrutura implementada 100% em código Object Pascal. As unidades FpImage.Reader.PNG, FpImage.Reader.JPEG e FpImage.Reader.Bitmap registram o suporte para a leitura de arquivos **PNG, JPEG e Bitmap**, respectivamente.





🔗 A PRIMEIRA TENTATIVA: 'HELLO, WORLD' (CONTINUAÇÃO)

Por fim, as unidades **FMX.Wasm.WindowHandle**, **FMX.Controls.Wasm**, **FMX.Canvas.Wasm** e **FMX.Platform.Wasm** ativam o back-end do **WebAssembly** para o **FMX**:

como os nomes das unidades sugerem, elas fornecem suporte a Window handle, estilo de controles e suporte a Canvas. A última unidade registra vários outros serviços de plataforma, embora muitos deles ainda estejam vazios no momento.

As unidades **FMX.Forms** e **forms.main** estão presentes em qualquer programa **FMX**: a primeira contém a implementação de **Application** e **TForm**, e a segunda contém a implementação do formulário principal do programa.

O formulário principal se parece com qualquer outro formulário **FMX**, com a exceção de que todos os controles são criados em código, em vez de depender do sistema de streaming que lê um form **.fmx**:

```
TForm1 = class(TForm)
  lblHello: TLabel;
  btnClickMe : TButton;
  procedure DoClick(Sender: TObject);
  procedure DoMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Single);
  procedure DoMouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Single);
  procedure HandlePaint(Sender: TObject; Canvas: TCanvas;
    const ARect: TRectF);
public
  constructor CreateNew(aOwner : TComponent; Dummy: NativeInt = 0); override;
end;
```

Como não estamos usando o sistema de streaming, o evento **varOnCreate** não pode ser usado para criar os controles, porque ele não é chamado.

Em vez disso, o construtor **CreateNew** deve ser usado. O construtor começa definindo as propriedades do formulário, a posição e o tamanho, bem como o título:

```
constructor TForm1.CreateNew(aOwner: TComponent; Dummy: NativeInt = 0);
begin
  inherited;
  Left := 0;
  Top := 0;
  Caption := 'FMX Hello world';
  Name := 'form1';
  Fill.Color := TAlphaColors.AquaMarine;
  Fill.Kind := TBrushKind.Solid;
  ClientHeight := 640;
  ClientWidth := 480;
  FormFactor.Width := 640;
  FormFactor.Height := 480;
  FormFactor.Devices := [TDeviceKind.Desktop];
  On-paint := HandlePaint;
```





🔗 THE FIRST TRY: 'HELLO, WORLD' (CONTINUATION)

Isso é o que pode ser encontrado em qualquer formulário `.fmx`, mas é traduzido para o código. A última linha implementa o manipulador `OnPaint` do formulário, que será usado para pintar o plano de fundo.

Depois de configurar o formulário, criamos um rótulo. Novamente, começamos definindo a posição, o tamanho e a legenda, tornando o rótulo visível;

```
lblHello := TLabel.Create (Self) ;  
With lblHello do  
begin  
    Font.Size := 12 ;  
    Font.Style := [] ;  
    Name := 'lblHello' ;  
    Parent := Self ;  
    Position.X := 80.0 ;  
    Position.Y := 56.0 ;  
    Size.Width := 241.0 ;  
    Size.Height := 41.0 ;  
    Size.PlatformDefault := False ;  
    Text := '"Hello, world!" from FMX' ;  
    TabOrder := 0 ;  
    HitTest := True ;  
    Visible := True ;  
    OnClick := DoClick ;  
    OnMouseUp := DoMouseUp ;  
    OnMouseDown := DoMouseDown ;  
end ;
```

As últimas linhas definem alguns manipuladores de eventos, simplesmente para demonstrar que nosso aplicativo reage aos eventos do usuário.

O último controle a ser criado é um botão.

```
btnClickMe := TButton.Create (Self) ;  
With btnClickMe do  
begin  
    btnClickMe.Font.Size := 12 ;  
    btnClickMe.Font.Style := [] ;  
    btnClickMe.Name := 'btnClickMe' ;  
    btnClickMe.Parent := Self ;  
    btnClickMe.Visible := True ;  
    btnClickMe.Position.X := 80.0 ;  
    btnClickMe.Position.Y := 150.0 ;  
    btnClickMe.Size.Width := 120.0 ;  
    btnClickMe.Size.Height := 32.0 ;  
    btnClickMe.Size.PlatformDefault := False ;  
    btnClickMe.Text := 'Click me... ' ;  
    btnClickMe.OnClick := DoClick ;  
end ;  
end ;
```





🔗 A PRIMEIRA TENTATIVA: 'HELLO, WORLD' (CONTINUAÇÃO)

É seguido o mesmo padrão dos controles anteriores: definição de posição, tamanho e legenda e, em seguida, o manipulador do evento `OnClick`. O manipulador de pintura define o pincel de preenchimento e desenha um retângulo:

```
procedure TForm1.HandlePaint(Sender: TObject; Canvas: TCanvas; const ARect: TRectF);  
begin  
    Canvas.Stroke.Color:=TAlphaColors.Yellow;  
    Canvas.Stroke.Kind:=TBrushKind.Solid;  
    Canvas.Fill.Kind:=TBrushKind.None;  
    Canvas.DrawRect(TRectF.create(79,55,80+242,56+42),1);  
end;
```

A cor de fundo do formulário é definida nos manipuladores `OnClick` do rótulo e do botão:

```
Procedure TForm1.DoClick(Sender : TObject);  
begin  
    Writeln('Got click :');  
    if Fill.Color=TAlphaColors.AquaMarine then  
        Fill.Color:=TAlphaColors.Lavender  
    else  
        Fill.Color:=TAlphaColors.AquaMarine  
    end;
```

Por fim, exibimos algumas mensagens quando o usuário clica com o mouse no rótulo:

```
Procedure TForm1.DoMouseUp(Sender: TObject; Button: TMouseButton;  
    Shift: TShiftState; X, Y: Single);  
begin  
    Writeln('Got mouse button "', Button, '" up at (' , X:6:2, ', ', Y:6:2, ')');  
end;  
Procedure TForm1.DoMouseDown(Sender: TObject; Button: TMouseButton;  
    Shift: TShiftState; X, Y: Single) begin  
    Writeln('Got mouse button "', Button, '" down at (' , X:6:2, ', ', Y:6:2, ')');  
end;
```

Com todo esse código, o aplicativo está pronto para ser compilado e executado.

O resultado pode ser visto na [figrefhelloworld](#) e, após um clique no botão,

o resultado pode ser visto na [figura 2](#) da página 15.

O botão não se parece com um botão normal do **Windows**.

Isso se deve ao fato de o arquivo de estilo ser do estilo **Linux ubuntu**:

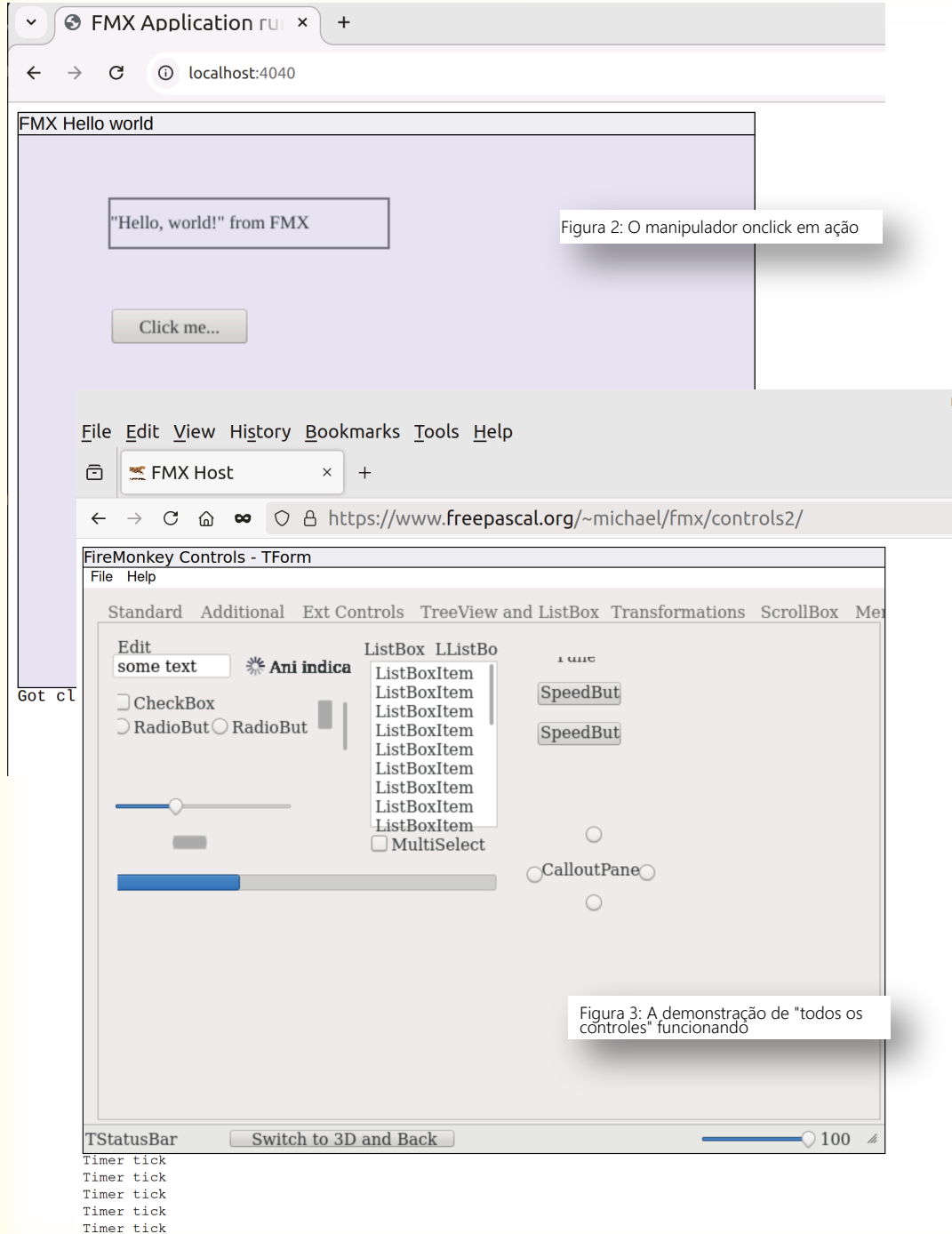
todo o desenvolvimento ocorre no **Linux** e, portanto, o estilo **ubuntu** foi escolhido.





COMPILAÇÃO DE UM APLICATIVO PARA O NAVEGADOR

ARTIGO
PÁGINA 16 / 26





8 INDO ALÉM

A demonstração "Todos os controles" do **FireMonkey** apresenta todos os controles possíveis do **FireMonkey**. É um bom teste para ver se todos os controles se comportam como esperado. Ela também mostrará que o carregamento do formulário funciona corretamente. No momento, são necessárias algumas modificações.

Uma das possibilidades do **FireMonkey** é usar a renderização em 3D. Isso obviamente requer suporte do backend e, no momento, esse suporte ainda não foi adicionado ao backend, embora esteja planejado para ser adicionado. Portanto, o botão para alternar para 3D deve ser removido (*ou, pelo menos, desativado e o código por trás dele removido*), bem como as unidades que se referem ao mecanismo de renderização 3D.

Uma vez feito isso, a demonstração é executada como planejado, com exceção dos componentes listview e treeview. Atualmente, a falha desses componentes ainda está sendo investigada. O resultado pode ser visto na figura 3 na página 16 deste artigo

9 DEBUGGING

O leitor atento deve ter notado que, ao criar a **RTL** e os pacotes, a opção -g foi especificada para o compilador. Isso significa que a **RTL** e os pacotes serão criados com informações de depuração. Também é gerada uma seção "names" que pode ser usada por todos os navegadores para exibir os nomes das funções.

Para poder usar as informações de depuração, você pode instalar um plug-in no navegador Chromium que permitirá depurar seu aplicativo **FMX** no navegador. O plug-in pode ser encontrado aqui

<https://chromewebstore.google.com/detail/cc++-devtools-support-dwa/pdcpmagijalfjmkmjngeonclgbbannb?Once>

Uma vez instalado, você tem muitas ferramentas disponíveis para analisar o que está acontecendo no seu programa: é possível colocar pontos de interrupção e percorrer o código, como pode ser visto na *figura 4 na página 17 e na figura 5 na página 18*.

Observe que o plug-in é voltado para **C/C++**, portanto, alguns tipos de informações ainda não estão disponíveis (*principalmente strings e propriedades*).

No entanto, ele torna a experiência de depuração muito melhor do que ter de escrever mensagens no console do navegador.

Para possibilitar uma maneira mais confortável de depurar, foi desenvolvido um módulo que permite inspecionar objetos usando uma interface mais do tipo do inspetor de objetos.

Esse módulo permite que você mostre uma árvore de objetos ativos e os inspecione.

Para usar esse instrumento, você precisa adicionar a unidade `wasm.debuginspector.rtti` à sua cláusula `uses`. Essa unidade está incluída nos pacotes **Free Pascal** para o compilador **WebAssembly**.



FMX Hello world

"Hello, world!" from FMX

Click me...

DEBUGGING (CONTINUATION)

Figure 4: The debugger in action

- top
 - localhost:4040
 - (index)
 - FMXHost.js
 - browser.dom.js
 - browser.min.js
 - hostconfig.js
 - oistyles.css
 - file://
 - home/michael/Projects/FM
 - Demos/FMX/HelloWorld
 - inc
 - HelloWorldFMX.dpr
 - form.main.pas
 - generics.collections.pa
 - generics.defaults.pas
 - rtti.pp
 - system.messaging.pp
 - Src/FMX



```

39   Writeln('Got mouse button "',Button,'" down at ('X:6:2,',',Y:6:2,')');
40   end;
41
42   Procedure TForm1.DoClick(Sender : TObject);
43
44   begin
45     Writeln('Got click :');
46     if Fill.Color=TAlphaColors.AquaMarine then
47       Fill.Color:=TAlphaColors.Lavender
48     else
49       Fill.Color:=TAlphaColors.AquaMarine
50   end;
51
52   procedure TForm1.HandlePaint(Sender: TObject; Canvas: TCanvas; const ARect:
53
54   begin
55     Canvas.Stroke.Color:=TAlphaColors.Yellow;
56     Canvas.Stroke.Kind:=TBrushKind.Solid;
57     Canvas.Fill.Kind:=TBrushKind.None;
58     Canvas.DrawRect(TRectF.create(79,55,80+242,56+42),1);
59   end;
60
61   constructor TForm1.CreateNew(aOwner: TComponent; Dummy: NativeInt = 0);
62   begin

```


Line 55, Column 16

- Hide network
- Preserve log
- Selected context only
- Group similar messages in console

Paused in debugger  






Objec

localhost:4040/

Debugger Performance insights 

Debugger interface showing a paused state. The main area displays a code editor with a yellow highlight on a line. The right sidebar contains a 'Paused on breakpoint' notification and a tree view of the current execution context. The tree view includes sections for Watch, Breakpoints, Scope, Parameter, Global, Call Stack, and the current function call stack.

- Paused on breakpoint
- Watch
- Breakpoints
 - Pause on uncaught exceptions
 - Pause on caught exceptions
- form.main.pas
 - Canvas.Stroke.Color... 55
- Scope
- Parameter
 - ▶ **ARECT**: TRECTF 
 - ▶ **CANVAS**: TCANVAS 
 - ▶ **SENDER**: TOBJECT 
 - ▶ **this**: TForm1 
- Global
 - ▶ **FORM1**: TForm1 
- Call Stack
 - ▶ HANDLEPAINT
 - form.main.pas:55

(From [HelloWorldFMX.wasm-0a45712e](#)) Coverage: n/a

Default levels  No Issues 

- Log XMLHttpRequests
- Eager evaluation
- Autocomplete from history
- Treat code evaluation as user action



🔍 DEBUGGING (CONTINUATION)

```
TWasmDebugInspector = Class(TComponent)
```

```
Public
```

```
  constructor Create(aOwner      : TComponent); override;
```

```
  destructor destroy; override;
```

```
  function ClearObjectTree      : Boolean;
```

```
  function ClearObjectInspector : Boolean;
```

```
  function SendObjectProperties(aObject: TObject; aVisibilities:TMemberVisibilities): Boolean;
```

```
  function SendObjectTree(aObject  : TObject; const aCaption : string): Boolean; virtual;
```

```
  function SendObjectTree(aObject: TObject): Boolean; virtual;
```

```
  Property OnGetObjectChildren : TObjectChildrenEvent;
```

```
end;
```

Figura 5: Informações úteis de depuração





COMPILAÇÃO DE UM APLICATIVO PARA O NAVEGADOR

ARTIGO PÁGINA 21 / 26

localhost:8090/

Recorder Performance insights

File Name));

Some call frames have warnings

Function Name	File Name
DESTROYHANDLE	FMX.Forms.pas:2998
DESTROYHANDLE	FMX.Forms.pas:6251
DESTROY	FMX.Forms.pas:2910
DESTROY	FMX.Forms.pas:6182
\$SYSTEM\$_\$TOBJECT_\$_\$\$_FREE	ControlsDemo.wa...721bba:0x503d4
MENUITEM6CLICK	ctrlsdemofrm.pas:399
HANDLEFRESNELMENUCLICK	FMX.Platform.Wasm.pas:1331
__FRESNEL_MENU_CLICK	fresnel.wasm.api.pp:526
rtl.module.rtl.createClass.HandleMenuClick	FMXHost.js:8116
rtl.module.rtl.createClass.DoMenuClick	FMXHost.js:6726
cb	FMXHost.js:257

Coverage: n/a





🕒 DEBUGGING (CONTINUAÇÃO)

Function WasmDebugInspector : TWasmDebugInspector;

A função **WasmDebugInspector** retorna uma instância pronta para uso dessa classe, mas você pode criar um descendente e usá-lo se quiser alterar seu comportamento. As duas chamadas importantes são

SendObjectTree que mostrará uma estrutura em árvore com um objeto no topo. Opcionalmente, você pode exibir uma legenda. Por padrão, ele seguirá a árvore de objetos de propriedade do Owner para determinar os filhos de um objeto, mas você pode alterar esse comportamento usando o manipulador de eventos `OnGetObjectChildren` (mais sobre isso abaixo).

SendObjectProperties exibirá as propriedades do objeto `aObject`, e você pode especificar as visibilidades que devem ser mostradas. (*está planejada uma extensão em que os campos também podem ser exibidos*)

As chamadas `ClearObjectTree` e `ClearObjectInspector` limparão as exibições da árvore de objetos e do inspetor de objetos.

Para poder mostrar a árvore ou o inspetor de objetos, é claro que você também deve adicionar algo ao programa host Javascript. O lado do host do objeto acima é implementado na classe `TWasmDebugInspectorApi`, implementada na unidade `debug.objectinspector.wasm.pas`, que você pode adicionar à cláusula `uses` do projeto, juntamente com as unidades `debug.objectinspector.html` e `wasm.debuginspector.shared`;

A classe `TWasmDebugInspectorApi` é apenas uma ponte entre o módulo **WebAssembly** e o ambiente **Javascript**. Ela usa duas classes para gerenciar a exibição real: `THTMLObjectTree` e `THTMLObjectInspector`. Esses dois objetos são responsáveis por exibir a árvore de objetos real e a grade do inspetor de objetos. Eles podem ser usados com o **PAS2JS** para exibir objetos **PAS2JS**, se você quiser. A árvore e a grade de propriedades devem ser exibidas em algum lugar da página HTML, portanto, adicionamos algumas tags HTML ao corpo da nossa página de hospedagem:

```
<div style="display: flex">
  <div id="desktop">
  </div>
  <div id="debug" style="display: flex">
    <div id="objectTree">
    </div>
    <div id="objectInspector">
    </div>
  </div>
</div>
```

Você pode ver que adicionamos um ID a cada um dos elementos relevantes, para poder acessar os elementos no código.

O inspetor de objetos e a árvore de objetos usam CSS, portanto, isso também deve ser incluído na página `index.html`.

<link href="oistyles.css" rel="stylesheet">

O construtor do nosso aplicativo sofreu algumas alterações: para tornar opcional a exibição da árvore de objetos e do inspetor de objetos, adicionamos um campo boolean `ShowOI` ao objeto de configuração do host e o usamos para mostrar ou ocultar o HTML:





🕒 DEBUGGING (CONTINUAÇÃO)

```
if Assigned(hostConfig) then
  begin
    WasiEnvironment.LogAPI := HostConfig.logWasiAPI;
    FFresnelApi.LogAPICalls := HostConfig.logFresnelAPI;
    WasiEnvironment.Environment.Add('FMX_LOGLEVEL=' + HostConfig.FMXLogLevel);
    DoShowOI := HostConfig.ShowOI;
  end
else
  DoShowOI := True;
if DoShowOI then
  ShowObjectInspector
else
  HideObjectInspector;
```

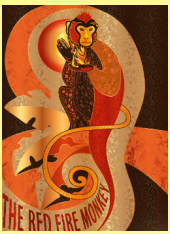
Onde as chamadas ShowObjectInspector e HideObjectInspector fazem o trabalho real. O HideObjectInspector é simples: ele altera o estilo de exibição da div na qual colocamos a árvore de objetos e o inspetor de objetos:

```
procedure TFMXHostApplication.HideObjectInspector;
begin
  GetHTMLElement('debug').style.setProperty('display', 'none');
end;
```

O ShowObjectInspector é um pouco mais complicado. Ele cria os dois renderizadores de html (THTMLObjectTree e THTMLObjectInspector) e define o ID do elemento em que eles devem renderizar a árvore e a grade:

```
procedure TFMXHostApplication.ShowObjectInspector;
begin
  FObjectTree := THTMLObjectTree.Create(Self);
  FObjectTree.ParentElementID := 'objectTree';
  FObjectInspector := THTMLObjectInspector.Create(Self);
  FObjectInspector.ParentElementID := 'objectInspector';
  FObjectInspector.Border := True;
  FObjectInspector.VisibleColumns := [{ocKind, ocVisibility}, ocName, ocValue];
  FObjectInspector.PropertyVisibilities := AllMemberVisibilities;
  FObjectInspectorAPI := TWasmObjectInspectorApi.Create(WasiEnvironment);
  FObjectInspectorAPI.DefaultInspector := FObjectInspector;
  FObjectInspectorAPI.DefaultObjectTree := FObjectTree;
  FObjectInspectorAPI.HandleInspectorEvents := [Low(THandleInspectorEvent),
                                                High(THandleInspectorEvent)];
  GetHTMLElement('debug').style.setProperty('display', 'flex');
end;
```





COMPILAÇÃO DE UM APLICATIVO PARA O NAVEGADOR

ARTIGO PÁGINA 24 / 26

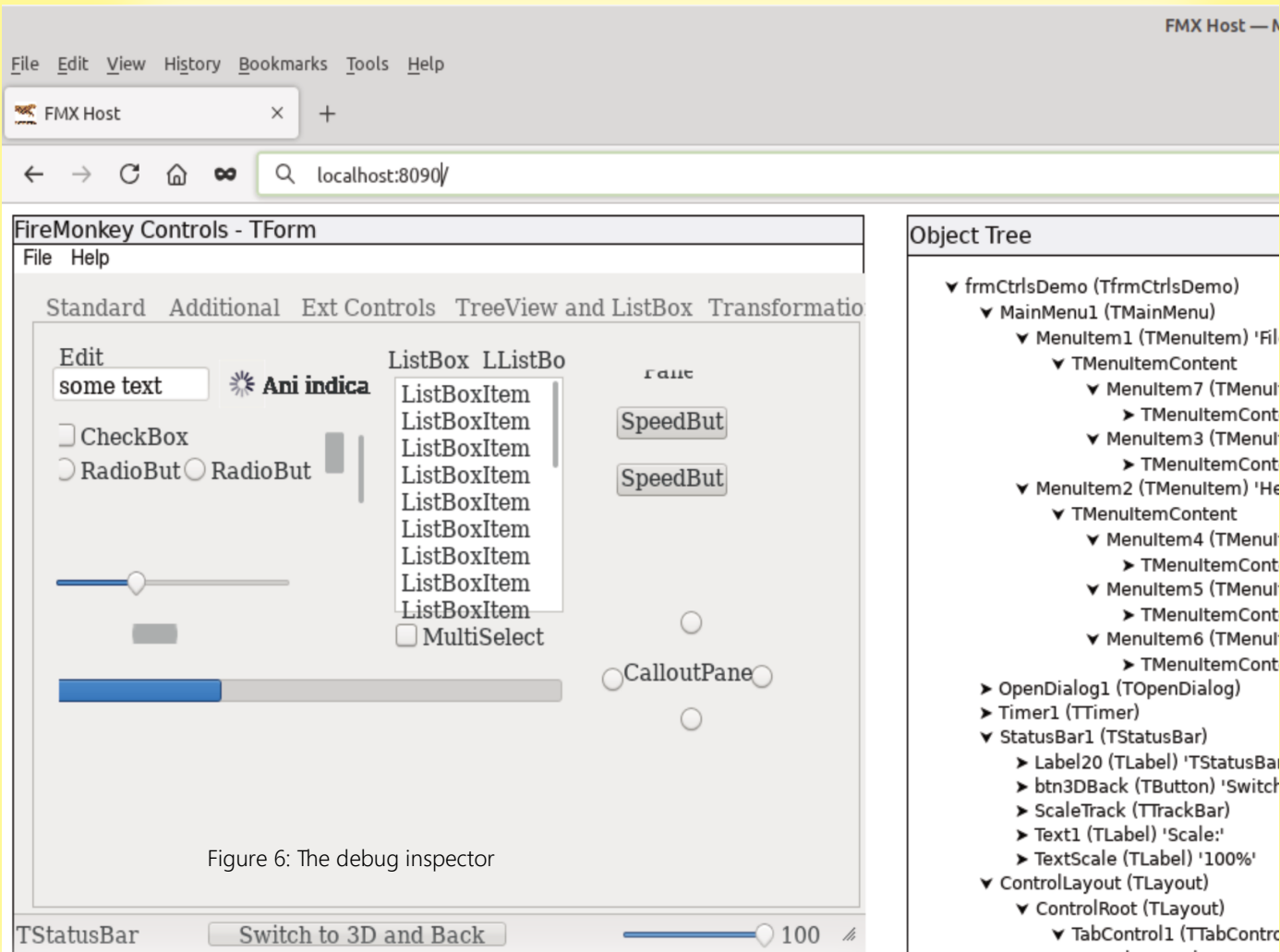


Figure 6: The debug inspector

DEBUGGING (CONTINUATION)

Por fim, a **API** é criada e conectada aos objetos de renderização HTML.

Observe que, com esse mecanismo, a **API** não estará disponível se você definir `ShowOI` como `false`. Como consequência, se o seu **WebAssembly** espera ter a **API** do inspetor de debug disponível, ele não será carregado. O código pode ser alterado com bastante facilidade para que a **API** esteja sempre disponível, mas a árvore e a grade do inspetor de propriedades simplesmente não são exibidas...

Além da árvore de objetos de propriedade do `Owner` definida pelo `TComponent`, o

FMX tem duas árvores de objetos pai-filho quando você cria um formulário e o preenche com controles. Todos os objetos gráficos descendem de `TFormObject`, que estão em uma árvore pai-filho. Mas nem todos os objetos gráficos são controles.

Os elementos de estilo são objetos gráficos, mas não são controles.

Portanto, os controles formam uma árvore separada dos objetos gráficos. A árvore mais interessante é a dos objetos gráficos, pois eles permitem que você examine também os elementos de estilo. Portanto, precisamos ter uma chamada de retorno para a nossa classe `TWasmDebugInspector`, que chamaremos de `DebugHelper`. Ela tem um método `DoChildren` que é usado como manipulador de eventos para o evento `TWasmDebugInspector.OnGetObjectChildren`.

RadioBu

Mozilla Firefox

→ Search

frmCtrlsDemo (TfrmCtrlsDemo)

Property Name	Value
Action	(empty)
ActionClient	False
Active	False
ActiveControl	(empty)
BiDiMode	bdLeftToRight
Border	TFormBorder ...
BorderIcons	(empty)
BorderStyle	Sizeable
Canvas	TWasmCanvas [00EBFFD0] (ID: 1) ...
Caption	FireMonkey Controls - TForm
Children	TFmxChildrenList ...
ChildrenCount	5
ClientHeight	540
ClientWidth	836
ComponentCount	269
ComponentIndex	0
ComponentState	(empty)
ComponentStyle	(empty)
Constraints	TSizeConstraints ...
Cursor	0
DesignInfo	0
Fill	TBrush ...
FormFactor	TFormFactor ...
FormFamily	TForm
FormState	(empty)
FormStyle	Normal
FullScreen	False
Handle	TWasmWindowHandle: (ID: 1) ...
Height	480
Index	-1
Left	497
ModalResult	0
Name	frmCtrlsDemo
Observers	TObservers ...
Owner	TApplication ...
Padding	TBoundsCommonCustomForm (0.00,0.00)-(0.00,0.00) ...
Parent	(empty)
ParentForm	(empty)

'Load Style...'

'Exit'

'Home'

'About...'

to 3D and Back'

(TTabControlContent)

'Standard'

leObject

yph

oStyleTextObject

oStyleObject

yph

oStyleTextObject

ent_ (TTabItemContent)

Label) 'ListBox & ListBoxItem'

Label) 'Edit'

1 (TEdit)

rowserStyledEdit

TStyledEditContent

ntentEdit

r1 (TScrollBar)

r2 (TScrollBar)

ox1 (TCheckBox) 'CheckBox'

utton1 (TRadioButton) 'RadioButton'

utton2 (TRadioButton) 'RadioButton'



9 DEBUGGING (CONTINUATION)

Os métodos `ShowObjectTree` mostram a árvore de objetos no console. Os métodos `SendObjectTree` enviam a árvore de objetos para o inspetor de objetos.

```
TDebugHelper = class
private
  FShowLog: Boolean;
  procedure DoChildren(aSender: TObject; aObject: TObject;
    var aChildren:TObjectDynArray;
    var procedure DoLog(Level: TWasmoILogLevel;
    const Msg: string);
  procedure DoObjectTree(ATree: TStrings; aObject: TfmXObject; const aPrefix:string);
Public
  constructor Create;
  destructor destroy; override;
  Procedure GetObjectTree(aTree : TStrings; aObject : TfmXObject);
  Procedure ShowObjectTree(aObject : TfmXObject; const aPrefix : String = '');
  Procedure SendObjectTree(aObject : TfmXObject);
  Procedure SendObjectTree(aObject : TfmXObject; const aCaption : string);
  Procedure SendObjectProperties(aObject : TfmXObject);
  property ShowLog : Boolean Read FShowLog Write FShowLog;
end;

var
  DebugHelper : TDebugHelper;

Procedure ShowObjectTree(aObject : TfmXObject; const aPrefix : String = '');
Procedure SendObjectTree(aObject : TfmXObject);
Procedure SendObjectTree(aObject : TfmXObject; aCaption : String);
Procedure SendObjectProperties(aObject : TfmXObject);
```

Com essa classe, a última coisa a fazer é mostrar a árvore de objetos e o objeto atual. Na classe de formulário `TfrmCtrlsDemo`, adicionamos as seguintes linhas ao construtor:

```
SendObjectTree(Self);
SendObjectProperties(Self);
```

O resultado é visível na figura 6 da página 21. Clicar em um objeto na árvore de objetos mostrará esse objeto no inspetor de objetos. Você pode usar o símbolo de reticências (...) para inspecionar as propriedades que são valores de classe: o inspetor de objetos mostrará o objeto ao qual a propriedade se refere. (*há uma seta para voltar*). Isso permite que você navegue entre os objetos e veja os valores digitados por string.

10 CONCLUSÃO

Embora ainda não esteja completa, graças ao backend do **Fresnel**, o porte do **FMX** para o navegador já está funcionando. Alguns controles ainda precisam de depuração e outros ainda precisam ser implementados. A ativação da renderização **3D** usando **WebGL** também está sendo planejada. O trabalho no **FMX** para a Web está progredindo, e o progresso será relatado aqui, mas aqueles que já desejam brincar com ele (*e possivelmente contribuir*) já podem fazê-lo.





Donate for Ukraine and get a free license at:

<https://components4developers.blog/2022/02/26/donate-to-ukraine-humanitarian-aid/>

If you are from Ukrainian origin you can get a free Subscription for Blaise Pascal Magazine, we will also give you a free pdf version of the Lazarus Handbook. You need to send us your Ukrainian Name and Ukrainian email address (*that still works for you*), so that it proofs you are real Ukrainian. please send it to editor@blaisepascal.eu and you will receive your book and subscription

BLAISE PASCAL MAGAZINE

Multi platform /Object Pascal / Internet / JavaScript / Web Assembly / Pas2Js /
Databases / CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux



Blaise Pascal



 **COMPONENTS
DEVELOPERS 4**

Donate for Ukraine and get a free license at:
<https://components4developers.blog/2022/02/26/donate-to-ukraine-humanitarian-aid/>

 **COMPONENTS
DEVELOPERS 4**





DONATE FOR UKRAINE AND GET A FREE LICENSE AT:

<https://components4developers.blog/2022/02/26/donate-to-ukraine-humanitarian-aid/>

kbmMW Professional and Enterprise NEW EDITION V. 5.23

kbmMemTable NEW EDITION V. 7.99.00

Standard and Professional Edition

kbmFMX Std/Pro v. 1.52.00 released

5.23.00 is a release with containing new stuff, refinements and bugfixes, **openssl v3 support**, WebSocket support, further improvements to SmartBind, new high performance hashing algorithms, improved RemoteDesktop sample and much more.

This release requires the use of **kbmMemTable** v. 7.98.00 or newer.

- RAD Alexandria supported
- Win32, Win64, Linux64, Android, IOS 32, IOS 64 and OS X client and server support
- Native high performance 100% developer defined application server
- Full support for centralised and distributed load balancing and fail-over
- Advanced ORM/OPF support including support of existing databases
- Advanced logging support
- Advanced configuration framework
- Advanced scheduling support for easy access to multi thread programming
- Advanced smart service and clients for very easy publication of functionality
- High quality random functions.
- High quality pronounceable password generators.
- High performance LZ4 and J peg compression
- Complete object notation framework including full support for YAML, BSON, Messagepack, J SON and XML
- Advanced object and value marshalling to and from YAML, BSON, Messagepack, JSON and XML
- High performance native TCP transport support
- High performance HTTPSys transport for Windows.
- CORS support in REST/HTML services.
- Native PHP, Java, OCX, ANSI C, C#, Apache Flex client support!

kbmMemTable is the fastest and most feature rich in memory table for Embarcadero products.

- Easily supports large datasets with millions of records
- Easy data streaming support
- Optional to use native SQL engine
- Supports nested transactions and undo
- Native and fast build in M/D, aggregation/grouping range selection features
- Advanced indexing features for extreme performance

- New: full Web-socket support.
- The next release of kbmMW Enterprise Edition will include several new things and improvements.
- One of them is full Web-socket support.
- New I18N context sensitive internationalisation framework to make your applications multilingual.
- New ORM LINQ support for Delete and Update.
- Comments support in YAML.
- New StreamSec TLS v4 support (by StreamSec)
- Many other feature improvements and fixes.

Please visit <http://www.components4developers.com> for more information about kbmMW

- High speed, unified database access (35+ supported database APIs) with connection pooling, metadata and data caching on all tiers
- Multi head access to the application server, via REST/AJAX, native binary, Publish/Subscribe, SOAP, XML, RTMP from web browsers, embedded devices, linked application servers, PCs, mobile devices, Java systems and many more clients
- Complete support for hosting FastCGI based applications (PHP/Ruby/Perl/Python, typically)
- Native complete AMQP 0.91 support (Advanced Message Queuing Protocol)
- Complete end 2 end secure brandable Remote Desktop with near realtime HD video, 8 monitor support, texture detection, compression and clipboard sharing.
- Bundling kbmMemTable Professional which is the fastest and most feature rich in memory table for Embarcadero products.

