# Manual de Usuario: uMakerAi.Prompts

## Introducción

La unidad ***uMakerAi.Prompts*** proporciona un componente Delphi para gestionar y usar plantillas de prompts (instrucciones) para interactuar con modelos de Inteligencia Artificial (IA). Facilita la creación, el almacenamiento y la personalización de prompts reutilizables, lo que simplifica el desarrollo de aplicaciones impulsadas por IA.

## Componentes Principales

La unidad ***uMakerAi.Prompts*** consta de dos clases principales:

* ***TAiPromptItem***: Representa un prompt individual.
* ***TAiPrompts***: Representa una colección de prompts.

## Uso del Componente TAiPrompts

1. **Agregar el Componente al Formulario:**

* Arrastra y suelta el componente TAiPrompts (desde la paleta de componentes) en tu formulario Delphi. Esto creará una instancia del componente, por ejemplo, AiPrompts1.

1. **Gestionar Prompts en Tiempo de Diseño:**

* Haz doble clic en el componente AiPrompts1 en el formulario. Esto abrirá el editor de colecciones de Delphi para la propiedad Items.
* Haz clic en el botón Add para crear un nuevo prompt (TAiPromptItem).
* **Nombre:** Establece la propiedad Nombre del prompt. Este nombre se utilizará para identificar el prompt en tu código.
* **Strings:** Haz clic en los puntos suspensivos (...) al lado de la propiedad Strings. Esto abrirá el editor de TStringList. Aquí puedes escribir el texto del prompt, línea por línea. El editor de TStringList permite editar prompts que abarcan varias líneas.

1. **Alternativa: Crear Prompts por Código:**

También puedes crear y agregar prompts a la colección Items por código:

var

PromptItem: TAiPromptItem;

begin

PromptItem := AiPrompts1.AddString('Saludo', 'Hola!');

PromptItem := AiPrompts1.AddString('SaludoPersonalizado', 'Hola ***<#Nombre>***, bienvenido ***a <#Ciudad>***!');

end;

## Métodos Principales del Componente TAiPrompts

### GetString(Nombre: String): String

* + Descripción: Recupera el texto de un prompt por su nombre.
  + Parámetros:
    - **Nombre**: El nombre del prompt que se va a recuperar.
  + Valor de Retorno: El texto del prompt (como una sola cadena, combinando las líneas del TStringList).
  + Ejemplo:

var

Saludo: string;

begin

Saludo := AiPrompts1.GetString('Saludo');

ShowMessage(Saludo); // Muestra "Hola!"

end;

### GetTemplate(Nombre: String; Params: Array of String): String; Overload

* + Descripción: Recupera el texto de un prompt y reemplaza marcadores de posición con valores proporcionados en un array de strings.
  + Parámetros:
    - **Nombre**: El nombre del prompt.
    - **Params**: Un array de strings en el formato 'Nombre=Valor'.
  + Valor de Retorno: El texto del prompt con los marcadores de posición reemplazados.
  + Ejemplo:

var

SaludoPersonalizado: string;

begin

SaludoPersonalizado := AiPrompts1.GetTemplate('SaludoPersonalizado', **['Nombre=Juan', 'Ciudad=Buenos Aires']**);

ShowMessage(SaludoPersonalizado); // Muestra "Hola Juan, bienvenido a Buenos Aires!"

end;

### GetTemplate(Nombre: String; Params: TStringList): String; Overload

* + Descripción: Recupera el texto de un prompt y reemplaza marcadores de posición con valores proporcionados en un TStringList.
  + Parámetros:
    - **Nombre**: El nombre del prompt.
    - **Params**: Un TStringList donde los nombres de las propiedades son los nombres de los marcadores de posición, y los valores de las propiedades son los valores de reemplazo.
  + Valor de Retorno: El texto del prompt con los marcadores de posición reemplazados.
  + Ejemplo:

var

SaludoPersonalizado: string;

Params: TStringList;

begin

Params := TStringList.Create;

try

Params.Values['Nombre'] := 'Juan';

Params.Values['Ciudad'] := 'Buenos Aires';

SaludoPersonalizado := AiPrompts1.GetTemplate('SaludoPersonalizado', Params);

ShowMessage(SaludoPersonalizado); // Muestra "Hola Juan, bienvenido a Buenos Aires!"

finally

Params.Free;

end;

end;

### GetTemplate(Nombre: String; Params: TJSONObject): String; Overload

* + Descripción: Recupera el texto de un prompt y reemplaza marcadores de posición con valores proporcionados en un TJSONObject.
  + Parámetros:
    - ***Nombre***: El nombre del prompt.
    - ***Params***: Un TJSONObject donde las claves son los nombres de los marcadores de posición, y los valores son los valores de reemplazo.
  + Valor de Retorno: El texto del prompt con los marcadores de posición reemplazados.
  + Ejemplo:

uses

System.JSON;

var

SaludoPersonalizado: string;

Params: TJSONObject;

begin

Params := TJSONObject.Create;

try

Params.AddPair('Nombre', 'Juan');

Params.AddPair('Ciudad', 'Buenos Aires');

SaludoPersonalizado := AiPrompts1.GetTemplate('SaludoPersonalizado', Params);

ShowMessage(SaludoPersonalizado); // Muestra "Hola Juan, bienvenido a Buenos Aires!"

finally

Params.Free;

end;

end;

### AddString(Nombre, Data: String): TAiPromptItem

* + Descripción: Agrega un nuevo prompt a la colección.
  + Parámetros:
    - ***Nombre***: El nombre del nuevo prompt.
    - ***Data***: El texto del nuevo prompt.
  + Valor de Retorno: Una referencia al TAiPromptItem creado.
  + Ejemplo:

var

NuevoPrompt: TAiPromptItem;

begin

NuevoPrompt := AiPrompts1.AddString('Despedida', 'Adiós!');

end;

## Sintaxis de los Marcadores de Posición

Los marcadores de posición en los prompts deben seguir el formato <#Nombre>, donde Nombre es el nombre del parámetro que se va a reemplazar.

**Ejemplo Completo**

unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, uMakerAi.Prompts, Vcl.StdCtrls;

type

TForm1 = class(TForm)

AiPrompts1: TAiPrompts;

Button1: TButton;

Memo1: TMemo;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var

Prompt: string;

Params: TStringList;

begin

// Obtener un prompt y reemplazar parámetros usando TStringList

Params := TStringList.Create;

try

Params.Values['Nombre'] := 'Juan';

Params.Values['Ciudad'] := 'Buenos Aires';

Prompt := AiPrompts1.GetTemplate('SaludoPersonalizado', Params);

Memo1.Text := Prompt;

finally

Params.Free;

end;

end;

end.

**Formulario (Unit1.dfm):**

1. Crea un nuevo formulario Delphi.
2. Arrastra y suelta un componente TAiPrompts desde la paleta de componentes. Nómbralo AiPrompts1.
3. Arrastra y suelta un componente TButton y un componente TMemo.

**Configuración del TAiPrompts (en tiempo de diseño):**

1. Haz doble clic en AiPrompts1.
2. Agrega un nuevo TAiPromptItem.
   * Nombre: SaludoPersonalizado
   * Strings:

Hola <#Nombre>, bienvenido a <#Ciudad>!

**Ejemplos:**

**Ejemplo 1: Video (Veo2)**

**Objetivo:** Generar un prompt para Veo2 que describa una escena realista en una cafetería.

**1.1. Creación del Prompt en Tiempo de Diseño:**

1. Haz doble clic en AiPrompts1.
2. Haz clic en "Add" para crear un nuevo TAiPromptItem.
   * Nombre: VideoCafeteria
   * Strings: (Haz clic en los puntos suspensivos para abrir el editor de TStringList)

Genera un video realista de:

Sujeto: <#Sujeto>

Contexto: <#Contexto>

Acción: <#Accion>

Estilo: Realista, con iluminación natural y colores vibrantes.

Movimiento de la cámara: Plano medio, siguiendo al sujeto mientras realiza la acción. Ligeros movimientos de cámara para mantener el interés visual.

Composición: Encuadre clásico, con el sujeto en el centro de la toma y el fondo ligeramente desenfocado.

Ambiente: Cálido y acogedor, con luz natural entrando por las ventanas y el sonido ambiental de la cafetería.

**1.2. Uso del Prompt en Delphi (Usando TStringList):**

uses

System.SysUtils, System.Classes;

procedure TForm1.ButtonVideoClick(Sender: TObject);

var

PromptFinal: string;

Params: TStringList;

begin

Params := TStringList.Create;

try

Params.Values['Sujeto'] := 'Una joven de unos 25 años, con cabello castaño y un vestido de verano floral.';

Params.Values['Contexto'] := 'Una cafetería concurrida en el centro de la ciudad durante la hora del almuerzo, con un día soleado.';

Params.Values['Accion'] := 'Trabajando en su computadora portátil y ocasionalmente mirando a su alrededor.';

PromptFinal := AiPrompts1.GetTemplate('VideoCafeteria', Params);

// Ahora PromptFinal contiene el prompt completo listo para enviar a Veo2.

MemoResultado.Text := PromptFinal; // Mostrar el prompt en un memo (opcional)

// (Aquí iría el código para enviar PromptFinal a la API de Veo2)

finally

Params.Free;

end;

end;

**1.3. Uso del Prompt en Delphi (Usando TJSONObject):**

uses

System.SysUtils, System.Classes, System.JSON;

procedure TForm1.ButtonVideoClick(Sender: TObject);

var

PromptFinal: string;

Params: TJSONObject;

begin

Params := TJSONObject.Create;

try

Params.AddPair('Sujeto', 'Una joven de unos 25 años, con cabello castaño y un vestido de verano floral.');

Params.AddPair('Contexto', 'Una cafetería concurrida en el centro de la ciudad durante la hora del almuerzo, con un día soleado.');

Params.AddPair('Accion', 'Trabajando en su computadora portátil y ocasionalmente mirando a su alrededor.');

PromptFinal := AiPrompts1.GetTemplate('VideoCafeteria', Params);

// Ahora PromptFinal contiene el prompt completo listo para enviar a Veo2.

MemoResultado.Text := PromptFinal; // Mostrar el prompt en un memo (opcional)

// (Aquí iría el código para enviar PromptFinal a la API de Veo2)

finally

Params.Free;

end;

end;

**1.4. Creación del Prompt por Código:**

uses

System.SysUtils, System.Classes;

procedure TForm1.FormCreate(Sender: TObject);

var

PromptItem: TAiPromptItem;

begin

PromptItem := AiPrompts1.AddString('VideoCafeteriaCode',

'Genera un video realista de:#13' +

'#13' +

'Sujeto: <#Sujeto>#13' +

'Contexto: <#Contexto>#13' +

'Acción: <#Accion>#13' +

'Estilo: Realista, con iluminación natural y colores vibrantes.#13' +

'#13' +

'Movimiento de la cámara: Plano medio, siguiendo al sujeto mientras realiza la acción. Ligeros movimientos de cámara para mantener el interés visual.#13' +

'Composición: Encuadre clásico, con el sujeto en el centro de la toma y el fondo ligeramente desenfocado.#13' +

'Ambiente: Cálido y acogedor, con luz natural entrando por las ventanas y el sonido ambiental de la cafetería.'

);

end;

y utilizas los ejemplos anteriores de **1.2. Uso del Prompt en Delphi (Usando TStringList):** o **1.3. Uso del Prompt en Delphi (Usando TJSONObject):** cambiando el nombre a VideoCafeteriaCode.

**Ejemplo 2: Imagen (Generación de Imágenes)**

**Objetivo:** Generar un prompt para un modelo de generación de imágenes (por ejemplo, DALL-E, Midjourney, Stable Diffusion) que cree una imagen de un gato en un jardín.

**2.1. Creación del Prompt en Tiempo de Diseño:**

1. Haz doble clic en AiPrompts1.
2. Haz clic en "Add" para crear un nuevo TAiPromptItem.
   * Nombre: ImagenGatoJardin
   * Strings:

Una fotografía de un gato <#Raza> durmiendo en un jardín lleno de <#Flores>, con luz <#Luz> y un estilo <#EstiloArtistico>.

**2.2. Uso del Prompt en Delphi (Usando Array of String):**

procedure TForm1.ButtonImagenClick(Sender: TObject);

var

PromptFinal: string;

begin

PromptFinal := AiPrompts1.GetTemplate('ImagenGatoJardin',

['Raza=Persa', 'Flores=Rosas rojas', 'Luz=Cálida', 'EstiloArtistico=Impresionista']);

// Ahora PromptFinal contiene el prompt completo listo para enviar a la API de generación de imágenes.

MemoResultado.Text := PromptFinal; // Mostrar el prompt en un memo (opcional)

// (Aquí iría el código para enviar PromptFinal a la API de generación de imágenes)

end;

**Ejemplo 3: Voz (Generación de Voz/Texto a Voz)**

**Objetivo:** Generar un prompt para un modelo de generación de voz o texto a voz que cree un mensaje de voz de bienvenida.

**3.1. Creación del Prompt en Tiempo de Diseño:**

1. Haz doble clic en AiPrompts1.
2. Haz clic en "Add" para crear un nuevo TAiPromptItem.
   * Nombre: VozBienvenida
   * Strings:

Genera un mensaje de voz de bienvenida para un nuevo usuario en la plataforma <#NombrePlataforma>, con un tono <#Tono> y una duración aproximada de <#Duracion> segundos.

**3.2. Uso del Prompt en Delphi (Usando TStringList):**

procedure TForm1.ButtonVozClick(Sender: TObject);

var

PromptFinal: string;

Params: TStringList;

begin

Params := TStringList.Create;

try

Params.Values['NombrePlataforma'] := 'MakerAI Suite';

Params.Values['Tono'] := 'Amable y entusiasta';

Params.Values['Duracion'] := '15';

PromptFinal := AiPrompts1.GetTemplate('VozBienvenida', Params);

// Ahora PromptFinal contiene el prompt completo listo para enviar a la API de generación de voz.

MemoResultado.Text := PromptFinal; // Mostrar el prompt en un memo (opcional)

// (Aquí iría el código para enviar PromptFinal a la API de generación de voz)

finally

Params.Free;

end;

end;

Estos ejemplos muestran cómo puedes utilizar el componente TAiPrompts para gestionar y personalizar prompts para diferentes tipos de modelos de IA (video, imagen y voz). Recuerda que estos son solo ejemplos básicos, y puedes adaptarlos y ampliarlos para satisfacer tus necesidades específicas.