

Almacenar datos en una base de datos local de SQLite.NET

En esta guía de inicio rápido se explica cómo almacenar datos en una base de datos de SQLite.NET local.

Realiza y documenta las siguientes actividades:

1. Inicia el entorno de desarrollo *Microsoft Visual Studio Community*.
2. En la ventana Inicio, haz clic en la solución *Notas* en la lista proyectos o soluciones recientes, o haz clic en abrir un proyecto o solución y, en el cuadro de diálogo Abrir proyecto o solución, selecciona el archivo de solución para el proyecto *Notas*.
3. Establece como proyecto de inicio el proyecto *Notes.Android*.
4. En Explorador de soluciones, selecciona el proyecto *Notes*, haz clic con el botón derecho y selecciona *Administrar paquetes NuGet...*
5. En el Administrador de paquetes NuGet, selecciona la pestaña *Examinar*, busca el paquete NuGet *sqlite-net-pcl*, selecciónalo y haz clic en el botón *Instalar* para agregarlo al proyecto.

Hay varios paquetes NuGet con nombres similares. El paquete correcto tiene estos atributos: Autor (s): Frank A. Krueger, Id.: *sqlite-net-pcl* y Vínculo de NuGet: *sqlite-net-pcl*.

Este paquete se usará para incorporar operaciones de bases de datos en la aplicación.

6. En Explorador de soluciones, en el proyecto *Notes*, abre el archivo *Note.cs* en la carpeta *Models* y reemplaza el código existente por el código siguiente:

```
using System;
using SQLite;

namespace Notes.Models {
    public class Note {
        [PrimaryKey, AutoIncrement]
        public int ID { get; set; }
        public string Text { get; set; }
        public DateTime Date { get; set; }
    }
}
```

7. Explica cada una de las líneas de código C#.
8. En Explorador de soluciones, agrega una nueva carpeta denominada *Data* al proyecto *Notes*.
9. En Explorador de soluciones, en el proyecto *Notas*, agrega una nueva clase denominada *NoteDatabase* a la carpeta de *Data*.
10. En el archivo *NoteDatabase.cs*, reemplaza el código existente por el código siguiente:

```
using System.Collections.Generic;
using System.Threading.Tasks;
using SQLite;
using Notes.Models;

namespace Notes.Data {
    public class NoteDatabase {
        readonly SQLiteAsyncConnection _database;

        public NoteDatabase(string dbPath) {
            _database = new SQLiteAsyncConnection(dbPath);
            _database.CreateTableAsync<Note>().Wait();
        }

        public Task<List<Note>> GetNotesAsync() {
```

```

        return _database.Table<Note>().ToListAsync();
    }

    public Task<Note> GetNoteAsync(int id) {
        return _database.Table<Note>()
            .Where(i => i.ID == id)
            .FirstOrDefaultAsync();
    }

    public Task<int> SaveNoteAsync(Note note) {
        if (note.ID != 0) {
            return _database.UpdateAsync(note);
        }
        else {
            return _database.InsertAsync(note);
        }
    }

    public Task<int> DeleteNoteAsync(Note note) {
        return _database.DeleteAsync(note);
    }
}
}

```

11. Explica cada una de las líneas de código C#.

12. En Explorador de soluciones, en el proyecto *Notes*, haz doble clic en el archivo *app.xaml.cs* para abrirlo. A continuación, reemplaza el código existente por el código siguiente:

```

using System;
using System.IO;
using Xamarin.Forms;
using Notes.Data;

namespace Notes {
    public partial class App : Application {
        static NoteDatabase database;

        public static NoteDatabase Database {
            get {
                if (database == null) {
                    database = new NoteDatabase(
                        Path.Combine(Environment.GetFolderPath(
                            Environment.SpecialFolder.LocalApplicationData),
                            "Notes.db3"));
                }
                return database;
            }
        }

        public App() {
            InitializeComponent();
            MainPage = new NavigationPage(new NotesPage());
        }

        protected override void OnStart() {
            // Handle when your app starts
        }

        protected override void OnSleep() {
            // Handle when your app sleeps
        }

        protected override void OnResume() {
            // Handle when your app resumes
        }
    }
}

```

```

    }
}

```

13. Explica cada una de las líneas de código C#.

14. En Explorador de soluciones, en el proyecto *Notes*, haz doble clic en el archivo *NotesPage.xaml.cs* para abrirlo. A continuación, reemplaza el método *OnAppearing* por el código siguiente:

```

protected override async void OnAppearing() {
    base.OnAppearing();

    listView.ItemsSource = await App.Database.GetNotesAsync();
}

```

15. Explica cada una de las líneas de código C#.

16. En Explorador de soluciones, haz doble clic en *NoteEntryPage.Xaml.cs* para abrirlo. A continuación, reemplaza los métodos *OnSaveButtonClicked* y *OnDeleteButtonClicked* por el código siguiente:

```

async void OnSaveButtonClicked(object sender, EventArgs e) {
    var note = (Note)BindingContext;
    note.Date = DateTime.UtcNow;
    await App.Database.SaveNoteAsync(note);
    await Navigation.PopAsync();
}

async void OnDeleteButtonClicked(object sender, EventArgs e) {
    var note = (Note)BindingContext;
    await App.Database.DeleteNoteAsync(note);
    await Navigation.PopAsync();
}

```

17. Explica cada una de las líneas de código C#.

18. En Visual Studio, selecciona el elemento de menú *Compilación > Compilar solución* (o pulsa F6). La solución se compilará y aparecerá un mensaje de operación correcta en la barra de estado de Visual Studio.

19. En la barra de herramientas de Visual Studio, pulsa el botón *Iniciar* (el botón triangular que parece un botón de reproducción) para iniciar la aplicación en la instancia elegida de Android Emulator.

20. Finaliza el dispositivo virtual creado.

21. Finaliza en entorno de desarrollo *Microsoft Visual Studio*.