

Documentación Reto técnico Mobile Developer Pleno at Versat.

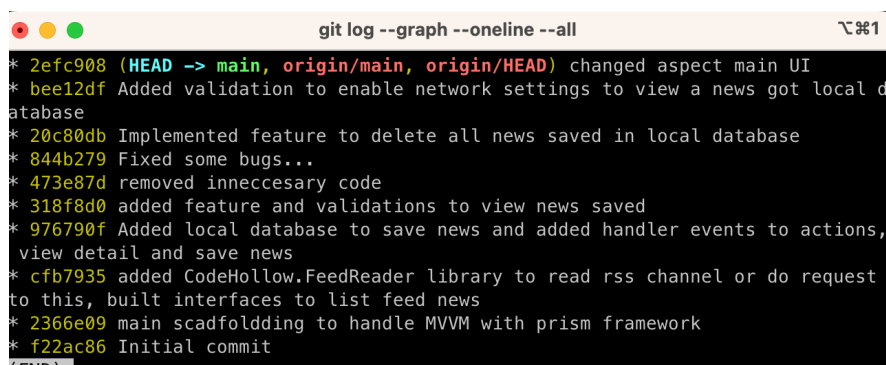
Repositorio, configuraciones y confirmaciones de proceso de construcción del reto:

Las URL publicas del repositorio del reto son las siguientes:

HTTPS: <https://github.com/rubberydev/NewsAppVersat.git>

SSH: <git@github.com:rubberydev/NewsAppVersat.git>

Muy importante revisar el historial de confirmaciones para tener una mejor comprensión de las actividades que realice y recursos que utilice para construir la App y así lograr tener fácilmente un mejor entendimiento de los componentes del sistema, en la siguiente imagen detallo el historial de confirmación organizadas desde la más reciente:



```
git log --graph --oneline --all
* 2efc908 (HEAD -> main, origin/main, origin/HEAD) changed aspect main UI
* bee12df Added validation to enable network settings to view a news got local database
* 20c80db Implemented feature to delete all news saved in local database
* 844b279 Fixed some bugs...
* 473e87d removed inneccesary code
* 318f8d0 added feature and validations to view news saved
* 976790f Added local database to save news and added handler events to actions, view detail and save news
* cfb7935 added CodeHollow.FeedReader library to read rss channel or do request to this, built interfaces to list feed news
* 2366e09 main scadfoldding to handle MVVM with prism framework
* f22ac86 Initial commit
```

Estos son los detalles del IDE que use como entorno de desarrollo: recordar que para probar la parte iOS se necesita obligatoriamente un computador Mac o una maquina virtual con el sistema operativo Mac OS en su última versión y el ultimo SDK de Xcode para todos los simuladores en la última versión de iOS16.2:

Visual Studio Community 2022 for Mac

Version 17.4.3 (build 21)

Installation UUID: c16186be-5c56-49c3-9822-ebb6267f6838

Runtime

.NET 6.0.12 (64-bit)

Architecture: X64

Roslyn (Language Service)

4.4.0-6.22578.12+3c6ab8e1715e5b080fb7bb77070810ab71e09387

NuGet

Version: 6.3.1.1

.NET SDK (x64)

SDK: /usr/local/share/dotnet/sdk/7.0.102/Sdks

SDK Versions:

7.0.102
6.0.405
5.0.408
5.0.407
5.0.403
5.0.402
5.0.401
5.0.400
5.0.302
5.0.201
3.1.420
3.1.418
3.1.415
3.1.414
3.1.413
3.1.412
3.1.411
3.1.407

MSBuild SDKs: /Applications/Visual

Studio.app/Contents/MonoBundle/MSBuild/Current/bin/Sdks

.NET Runtime (x64)

Runtime: /usr/local/share/dotnet/dotnet

Runtime Versions:

7.0.2
6.0.13
5.0.17
5.0.16
5.0.12
5.0.11
5.0.10
5.0.9
5.0.8
5.0.4
3.1.26
3.1.24
3.1.21
3.1.20

3.1.19
3.1.18
3.1.17
3.1.13

Updater
Version: 11

Apple Developer Tools
Xcode 14.2 (21534)
Build 14C18

Xamarin.Mac
Version: 9.0.0.27 (Visual Studio Community)
Hash: 933c6c2c9
Branch: xcode14.1
Build date: 2022-11-22 02:00:36-0500

Xamarin.iOS
Version: 16.1.1.27 (Visual Studio Community)
Hash: 933c6c2c9
Branch: xcode14.1
Build date: 2022-11-22 02:00:37-0500

Xamarin Designer
Version: 17.4.0.136
Hash: d49c9ff6d3
Branch: remotes/origin/d17-4
Build date: 2023-01-12 19:21:10 UTC

Xamarin.Android
Version: 13.1.0.1 (Visual Studio Community)
Commit: xamarin-android/d17-4/13ba222
Android SDK: /Users/julian/Library/Developer/Xamarin/android-sdk-macosx
Supported Android versions:
 11.0 (API level 30)
 10.0 (API level 29)
 7.1 (API level 25)
 9.0 (API level 28)
 13.0 (API level 33)
 7.0 (API level 24)

SDK Command-line Tools Version: 7.0

SDK Platform Tools Version: 33.0.2

SDK Build Tools Version: 32.0.0

Build Information:

Mono: a96bde9

Java.Interop: xamarin/java.interop/d17-4@fcc33ce2

SQLite: xamarin/sqlite/3.39.3@23e1ae7

Xamarin.Android Tools: xamarin/xamarin-android-tools/main@0be567a

Microsoft Build of OpenJDK

Java SDK: /Library/Java/JavaVirtualMachines/microsoft-11.jdk

11.0.12

Android Designer EPL code available here:

<https://github.com/xamarin/AndroidDesigner.EPL>

Eclipse Temurin JDK

Java SDK: /Library/Java/JavaVirtualMachines/temurin-8.jdk

1.8.0.302

Android Designer EPL code available here:

<https://github.com/xamarin/AndroidDesigner.EPL>

Android SDK Manager

Version: 17.4.0.54

Hash: 6eabb9e

Branch: remotes/origin/d17-4

Build date: 2023-01-12 19:21:16 UTC

Android Device Manager

Version: 0.0.0.1206

Hash: 886af39

Branch: 886af39

Build date: 2023-01-12 19:21:16 UTC

Build Information

Release ID: 1704030021

Git revision: 26eac6764d85f12fe50ee0d45a7cd0b266be7b23

Build date: 2023-01-12 19:19:02+00

Build branch: release-17.4

Build lane: release-17.4

Operating System

Mac OS X 13.1.0

Darwin 22.2.0 Darwin Kernel Version 22.2.0

Fri Nov 11 02:08:47 PST 2022

root:xnu-8792.61.2~4/RELEASE_X86_64 x86_64







Descripción de flujos, librerías utilizadas, UX(experiencia de usuario) GUI(Interfaz Gráfica de Usuario) y casos de uso:

IMPORTANTE:

- la librería usada para usar MVVM es recomendada por uno de los dos creadores del compilador de Xamarin(Mono Runtime para poder ser ejecutado en diferentes entornos) el cual es una alternativa al CLR(Common Language Runtime clásico para entornos Windows):



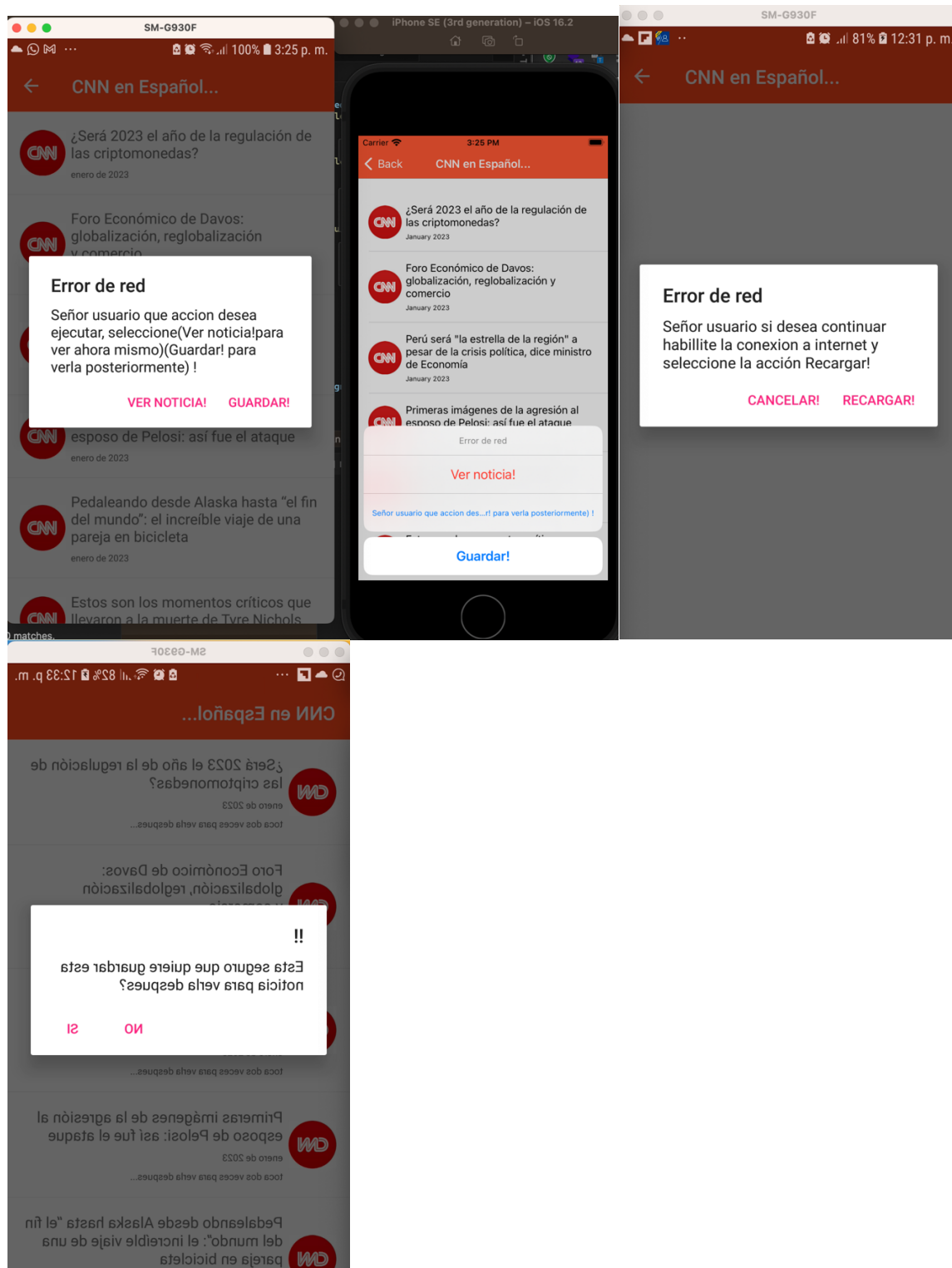
- Las interfaces que están creadas con programación imperativa en C#, las hice así, con el fin de no reinventar la rueda, ya que podía reutilizar un código de la comunidad open source q hicimos hace unos años cuando Xaml no estaba muy maduro en XF, y también teniendo en cuenta que Xamarin maneja estas filosofías open source y de software libre.
- Para la construcción de la aplicación multiplataforma use Xamarin.Forms en su versión más reciente y así hacer disponible la aplicación en Android y iOS,
- CodeHollow.FeedReader para recuperar los datos o hacer peticiones de la fuente RSS configurada .NetStandard como librería de arquitectura base para trabajar con los SDK de Xamarin y aprovechar las utilidades de C# language y .net framework, la librería Prism para trabajar con el patrón de diseño MVVM también use el patrón singleton para manejar una única instancia de un objeto durante todo el ciclo de vida de la app y poder hacer algunas validaciones importantes, además use DI(inyección de dependencias con la librería Prism para dar más seguridad en la capa de acceso a datos en local y la navegabilidad, para persistir datos en local use SQLite para .net framework que es una biblioteca de clases q me permite persistir grandes volúmenes de datos con .net framework en las plataformas respectivas, Xamarin.Essentials para acceder a las API del dispositivo respectivamente:

| | |
|---|---|
|  | CodeHollow.FeedReader 1.2.4 |
| | RSS and ATOM feed reader/parser. FeedReader is a .net standard 1.3 library used for reading and parsing RSS and ATOM feeds. Supports RSS 0.91, 0.92, 1.0, 2.0 and ATOM. Library tested with multiple |
|  | NETStandard.Library 2.0.3 |
| | A set of standard .NET APIs that are prescribed to be used and supported together. 18a36291e48808fa7ef2d00a764ceb1ec95645a5 When using NuGet 3.x this package requires at least version 3.4. |
|  | Prism.Unity.Forms 8.1.97 |
| | Use these extensions to build Xamarin.Forms applications with Prism and Unity. |
|  | sqlite-net-pcl 1.8.116 |
| | SQLite-net is an open source and light weight library providing easy SQLite database storage for .NET, Mono, and Xamarin applications. This version uses SQLitePCLRaw to provide platform independent |
|  | Xamarin.Essentials 1.7.4 |
| | Xamarin.Essentials: a kit of essential API's for your apps |
|  | Xamarin.Forms 5.0.0.2545 |
| | Build native UIs for iOS, Android, UWP, macOS, Tizen and many more from a single, shared C# codebase |

Las tres opciones que le brindo al usuario se encuentran en la vista principal que carga cuando inicia la app, El **botón Azul** le permite al usuario cargar las noticias desde el canal RSS respectivo, fuente de las noticias:

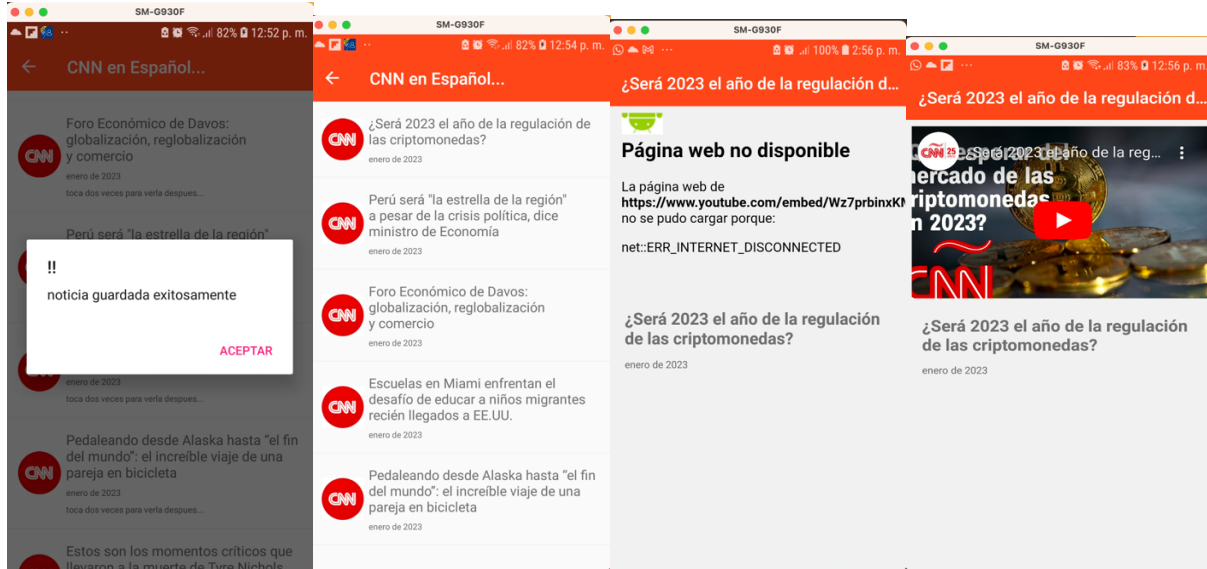


Una vez se accede al botón se puede visualizar cada uno de los ítems (**tal como se ve en la anterior imagen**) o noticias recuperadas de la fuente RSS configurada, se debe tener conexión a internet o si no aparece un mensaje que le indica al que active las opciones de internet para continuar y luego seleccione recargar, cuando el usuario hace tap en un ítem el sistema le pregunta que acción desea ejecutar como lo muestran las siguientes imágenes:



Si se guarda exitosamente la noticia, la app lo confirma, y si posteriormente con el **botón rojo** se accede a ver las noticias guardadas, solo permite ver el detalle y debe tener activas las configuraciones de red porque si no, no puede cargar el video de la noticia, también se puede

observar que no está el icono de flecha atrás en el navbar, ya que lo deshabilite para forzar al usuario q use el botón físico del dispositivo o virtual del sistema operativo y así yo poder responder al evento y llevarlo a la vista principal con las tres acciones o botones y siempre se cargue la información actualizada de la fuente de datos (**sea el canal rss o la base de datos local**), la app es orientada a conexión por lo que se debe tener siempre activada la config de internet para su correcto funcionamiento:



Aunque el reto no me lo solicitaba agregue un **botón naranja** para borrar todas las noticias en local, cada vez que borra una adecuadamente aparece el mensaje o el detalle de la excepción controlada adecuadamente en caso de suceder, además si trato de ver noticias en local el sistema también me informara que no hay noticias guardadas, si todas se borraron correctamente:

