



KD-40540

Build Android Apps with C#
and .NET using the Xamarin
Tools for Visual Studio

www.ked.com.mx

Av. Revolución No. 374 Col. San Pedro de los Pinos, C.P. 03800, México, CDMX. Tel.: 55 52 78 55 60





About this course.

This course takes experienced C# .NET developers through the entire Xamarin.Android mobile development process, using the Xamarin tools for Visual Studio. Topics range from the basics such as Android activities, intents, and fragments and then move to more advanced topics such as consuming REST-based services, ListViews, data storage with SQLite and publishing to the Google Play store.

Length.

5 Days.

Audience profile.

This course is intended for developers with at least basic C# knowledge with 6-12 months of .NET experience. Students should own and/or have used an Android device and be familiar with the paradigms of the platform.

Hardware/Software:

- A Mac running on the latest OS, with Visual Studio for Mac installed or a Windows 10 PC with Visual Studio 2017 and the Mobile development with .NET workload installed.
- An Android device is highly recommended.

Prerequisites.

Before attending this course, students must have the following technical knowledge:

- Basic C# knowledge with 6-12 months of .NET experience.

At course completion.

After completing this course, students will be able to:

- Build Android applications in C# using the Xamarin tools for Visual Studio.
- Design multiscreen applications using native Android controls and AXML layouts.
- Access REST services, store data locally using a SQLite database, and customize the data presentation using native controls and custom listview cells.

Exam.

None.

Course outline.

Module 1: Getting Started with Xamarin (XAM101).

By the end of this module, you will be able to choose the Xamarin approach that is right for you. You will also set up your development machine and run an app on Windows, the iOS simulator, and an Android emulator.

- Define the state of mobile technology today.
- Discuss mobile app trends.
- Identify approaches to development.
- Discover the Xamarin Approach.
- Set up your development environment.

Labs: Building Cross Platform Applications with Xamarin.

- Demonstration - Cross Platform Applications with Xamarin.

- Demonstration – View a Xamarin.Forms Project.
- Setup Validation – Validate your development environment ready.

Module 2: Introduction to Xamarin.Android (AND101).

During this module you will create a new project in Visual Studio, code the user interface and behavior for an Activity, build the project, and test it on an Android emulator. You'll also see how to keep your development environment up-to-date as new Android versions are released.

- Create a Xamarin.Android project.
- Decompose an app into Activities.
- Build an Activity's UI.
- Write an Activity's behavior.
- Update your Android SDK.

Labs: Create a Xamarin.Android application with Views, layouts and behavior.

- Create a Xamarin.Android Project.
- Add views to a layout file manually and with the Designer.
- Implement an Activity's behavior and run your app in an emulator.
- Update the Tools and SDK Platform.

Module 3: Activities and Intents (AND102).

In this module you will code the core patterns that make Activities work together. You will see how to start Activities, pass arguments, and retrieve results. You will also see how to launch standard Android Activities such as the camera or the web browser.

- Start an Activity in your .apk.
- Finish an Activity.
- Pass arguments to an Activity.
- Get Activity results.
- Start a system Activity.

Labs: Create an app with collaborating Activities.

- Explore the completed lab exercise.
- Start an Activity in your .spk.
- Programmatically end an Activity.
- Pass arguments to an Activity.
- Get Activity Results.
- Launch a system Activity.

Module 4: List Views and Adapters in Android (AND110).

In this module you will code an Adapter to populate the ListView and handle item-selection events to implement a simple master-detail pattern. You will also see two advanced tasks: layout recycling with view holders to boost performance and fast scrolling with section indexers to help the user navigate large data sets.

- Handle list-item click events.
- Implement a custom adapter.
- Use layout recycling and the view-holder pattern.

**Labs: Use an Adapter to populate a ListView and then customize it to boost efficiency.**

- Populate a ListView using an ArrayAdapter.
- Handle list-item click events.
- Implement a custom adapter.
- Use layout recycling and the view-holder pattern.
- Enable fast scrolling and a code a section indexer.

Module 5: RecyclerView and CardView (AND115).

User RecyclerView to display a data collection and CardView to host the visualization of each element.

- Display a collection using RecyclerView.
- Update the UI after a data change.
- Respond to user actions.
- Show data in a CardView.

Module 6: Consuming REST-based Web Services (XAM150).

In this module, you will learn to consume REST-based web services with HttpClient using both the managed networking stack and the platform-specific handlers that boost performance. You will also learn some common strategies for dealing with the unique challenges that mobile devices face when communicating over the network.

- Obtain the device's network capabilities.
- Introduce REST.
- Consume REST services with Xamarin.

Labs: Use, consume and communicate with REST services.

- Determine the network connectivity.
- Communicate with a Book Service.
- Demonstration: Leverage the native platform network stack.

Module 7: Working with SQLite and Mobile Data (XAM160).

During this module you will learn to identify the proper location for your database file and how to insert, update, retrieve, and delete data efficiently using asynchronous I/O calls.

- Choose a data storage strategy.
- Store data locally with SQLite.
- SQLite asynchronously.

Labs: Utilize SQLite to store and access data.

- Determine your database file path.
- Add SQLite.Net to your projects.
- Access a SQLite database with SQLite.Net.
- Access a SQLite database using asynchronous methods.

Module 8: Toolbar and App Bar (AND180).

In this module you will learn to create a Toolbar, populate it from an XML menu file, and respond to item-click events. You will also learn how to integrate a Toolbar into your Activity by setting it as the Activity's app bar.

- Add a Toolbar to an Activity.
- Populate Toolbar Actions from an XML menu file.
- Use a Toolbar as an Activity's app bar.
- Set a navigation icon on an app bar.

Labs: Create and populate a Toolbar and set it as the Activity's AppBar.

- Add a Toolbar to an Activity.
- Populate Toolbar Actions from an XML menu file.
- Add a checkable submenu.
- Use a Toolbar as an Activity's app bar.
- Set a navigation icon on an app bar.

Module 9: Android Navigation (AND205).

This module teaches you how to code three common navigation patterns: tab, gestural, and drawer. It also includes guidelines to help you decide which pattern is appropriate for your app as well as teaches you to use fragments and FrameLayout.

- Use Fragment and FrameLayout to swap views.
- Implement tab navigation with TabLayout.
- Implement gestural navigation with ViewPager.
- Combine gestural and tab navigation.
- Implement drawer navigation with DrawerLayout.

Labs: Build three common navigation patterns in Xamarin.Android.

- Use Fragment and FrameLayout to swap views.
- Implement tab navigation with TabLayout.
- Implement gestural navigation with ViewPager.
- Combine gestural and tab navigation.
- Implement drawer navigation with DrawerLayout.

Module 10: Android Services (AND210).

This module teaches you to define a Service, use start/stop methods to explicitly control its lifecycle, and use bind/unbind methods to automatically destroy the instance when the last client has unbound. Perform long-running operations using Android services.

- Handle list-item click events.
- Implement a custom adapter.
- Use layout recycling and the view-holder pattern.

Labs: Use an Adapter to populate a ListView and then customize it to boost efficiency.

- Populate a ListView using an ArrayAdapter.
- Handle list-item click events.
- Implement a custom adapter.
- Use layout recycling and the view-holder pattern.
- Enable fast scrolling and a code a section indexer.



Module 11: Google Maps (AND230).

During this module will learn how to use the Google Play Services APIs to add a map to your app. You will perform the entire setup process: register your app in the Google APIs Console, create an API key, and add your API key to your app's manifest. You will then use MapView and MapFragment to visualize map data in your app's UI. Finally, you will write code to add markers to the maps and adjust the viewpoint position and map perspective.

- Choose a map display strategy.
- Configure an application to use Google Maps.
- Display a map in your application.
- Add markers to a Google Map.
- Adjust the map position and perspective.

Labs: Create an app that shows a Google Map with markers and viewpoints.

- Group Exercise: Locate your keystore and SHA1 fingerprint.
- Group Exercise: Obtain a Google API key for Google Maps V2.
- Add support for Google Maps.
- Add markers to your map.
- Change your map viewpoint.

Module 12: Touch in Android (AND205).

This module teaches you how to code three common navigation patterns: tab, gestural, and drawer. It also includes guidelines to help you identify which pattern is most appropriate for your app.

- Use Fragment and FrameLayout to swap views.
- Implement tab navigation with TabLayout.
- Implement gestural navigation with ViewPager.
- Combine gestural and tab navigation.
- Implement drawer navigation with DrawerLayout.

Labs: Build three common navigation patterns in Xamarin.Android.

- Use Fragment and FrameLayout to swap views.
- Implement tab navigation with TabLayout.
- Implement gestural navigation with ViewPager.
- Combine gestural and tab navigation.
- Implement drawer navigation with DrawerLayout.

Module 13: Gestures (AND231).

In this module you use Android's gesture recognizers to respond to user gestures. You will learn the full list of standard Android gestures and see how to write your own recognizer to support a custom rotation gesture.

- Determine the current location.
- Search for points of interest around the device.

Labs: Implementing and managing Azure AD synchronization.

- Group Exercise: Setup the Google Maps API key.
- Working with the Location Manager.
- Identifying a provider based on criteria.



- Group Exercise: Adding support for LocationServices.
- Using Geocoder to perform reverse geocoding.
- Using Geocoder to find specific POI.

Module 14: Performance Tuning (XAM370).

During this module you will learn how memory leaks happen in managed memory even with a sophisticated garbage collector and how to discover and fix them. You will then learn several memory issues that are specific to Xamarin.iOS and Xamarin.Android.

- Identify and fix memory leaks in your code.
- Recognize and fix Xamarin.iOS specific memory problems.
- Recognize and fix Xamarin.Android specific memory problems.

Labs: Track down and correct memory leaks in your Xamarin.Forms app.

- Using the Xamarin Profiler to monitor allocations.
- Finding and fixing delegate reference leaks.
- Identifying and breaking strong reference cycles.
- Watch out for peer promotions.
- Show ListView memory and performance with a custom adapter.

Module 15: Preparing for Publishing (XAM220).

Prepare your app for release through the App Store, Google Play, an Windows Marketplace.

- Getting ready to publish your app.
- Understanding publishing styles.
- Publishing to a store.

Labs: Implementing and managing Azure AD synchronization.

- Update your app for publishing.
- Choose a distribution strategy.
- Publish to a store.

Module 16: Android Publishing (AND220).

Build an app package and submit it to the Google Play Store and Amazon App Store.

- Configure your app for publishing.
- Create a Google Play developer account.
- Create an app package.
- Submit your application for publishing.

Labs: Implementing and managing Azure AD synchronization.

- Configuring directory synchronization.
- Synchronization directories.

clientes@ked.com.mx

Av. Revolución No. 374 Col. San Pedro de los Pinos, C.P. 03800, México, CDMX. Tel.: 55 52 78 55 60

