Cuestionario Mobile

**Integrantes del grupo:**

* Goncalves, Lucas
* Jorreto, Mariano
* López, Ariel
* Schiter, Tomás
* Zurriable, Jorge

1. **¿Cuáles son los protocolos de mobile?**

WAP es el protocolo a través del cual se realiza la comunicación..

WAP1 con WML (Wireless Markup Language)

WAP2 con XHTML (eXtensibleHTML: como HTML pero con sintáxis XML)

cHTML (compact HTML) es una versión reducida de HTML 3.2 que se usa en mobile.

HTML usa JavaScript como lenguaje cliente.

WML usa WMLScript como lenguaje cliente que se ejecuta directamente en el celular.

1. **Clasifique concretamente dispositivos móviles.**

Personal Digital Assistants (PDAs)

* Caben en una mano.
* Tienen su pantalla grande en proporción con el tamaño del equipo.
* Pantalla touch-screen a través de un stylus (lápiz).
* Carecen de teclado (generalmente).
* Comunicación por infrarrojo. Bluetooth o cable.
* Acceso a Internet.
* Sistema Operativo con soft utilitario que permite instalar aplicaciones.

Dos tipos de PDAs:

**PalmOS**

Son los equipos que usan el sistema operativo PalmOS.

**PocketPC**

Solución de Microsoft.

Ofrecen un Sistema Operativo Windows reducido.

Son equipos tipo agenda electrónica que se abren y poseen pantalla y teclado.

Small Personal Object Tecnology (SPOTs)

Son aplicaciones inteligentes de uso personal y para el hogar, productos electrónicos de consumo y otros objetos basados en nuevas capacidades de hardware y software.

Fue iniciado por Microsoft utilizando relojes con esta nueva tecnología.

Tarjetas Inteligentes

Son SIM Cards de teléfonos celulares GSM con tecnología Java Card.

Teléfonos Celulares

Plataformas Propietarias

Son la mayoría de los teléfonos celulares: Sony Ericsson, Samsung, Siemens, Motorola, Nokia, Windows Smartphone.

1. **¿Qué es una solución stand-alone?**

Son aquellas que se desarrollan para ser instaladas y ejecutadas sobre el equipo móvil y que funcionan en forma desconectada de Internet o de un servidor central.

Para su desarrollo debe generarse un paquete ejecutable con un formato compatible para el sistema operativo donde será instalado.

1. **¿Qué es una solución online (conectada)?**

Son soluciones a través de Internet usando páginas web o WAP para la interfaz. Toda la ejecución la realizan en el servidor. Se puede desarrollar en diferentes lenguajes que son comprensibles por varios dispositivos (HTML, WML, cHTML, etc.)

1. **Establezca la diferencia entre stand-alone y online. Ventajas y desventajas.**

El Stand-alone es más veloz y con mejor manejo de memoria. Este permite trabajar offline y posee soporte de sincronización con un equipo de escritorio. Sus desventajas son la necesidad de desarrollar distintas versiones para sistemas operativos, la imposibilidad de consultar centros de datos remotos y que soportan poca información.

El online ofrece mayor compatibilidad que el stand-alone y tiene como grandes ventajas la mayor cantidad de información que soporta, la capacidad de hacer operaciones complejas en el servidor y que no se necesita instalar ninguna aplicación. Sus desventajas son la necesidad de estar siempre online y su lentitud en comparación con la solución online.

1. **¿Qué es una solución smart client?**

Son la solución que combina lo mejor de las stand-alone y las online.

Son aplicaciones ejecutables que se instalan en los equipos y usan la conexión para comunicarse e interactuar con un servidor.

Esto permite trabajar con una gran cantidad de información y trabajar offline.

Su desventaja es que es muy complejo de desarrollar, que debe crearse el cliente basándose en todos los tipos de sistemas operativos y que debe distribuirse e instalarse en todos los equipos.

1. **Cuando se trabaja con aplicaciones que se ejecutan directamente sobre el equipo distinguimos dos tipos de código para generar: código nativo y código manejado. Describa y compare ventajas y desventajas de los dos anteriores.**

Código nativo

Este tipo de código implica que el archivo ejecutable a instalar en el equipo está expresado con código ensamblador entendible por el sistema operativo y por el procesador del equipo.

Sus ventajas son la rapidez de ejecución, el acceso a todas las capacidades del equipo y que no requiere instalaciones adicionales para ejecutarse.

Sus desventajas son la necesidad de compilar el proyecto en cada sistema operativo o hardware para luego generar el ejecutable y que acceder directamente a la memoria puede traer problemas.

Código manejado

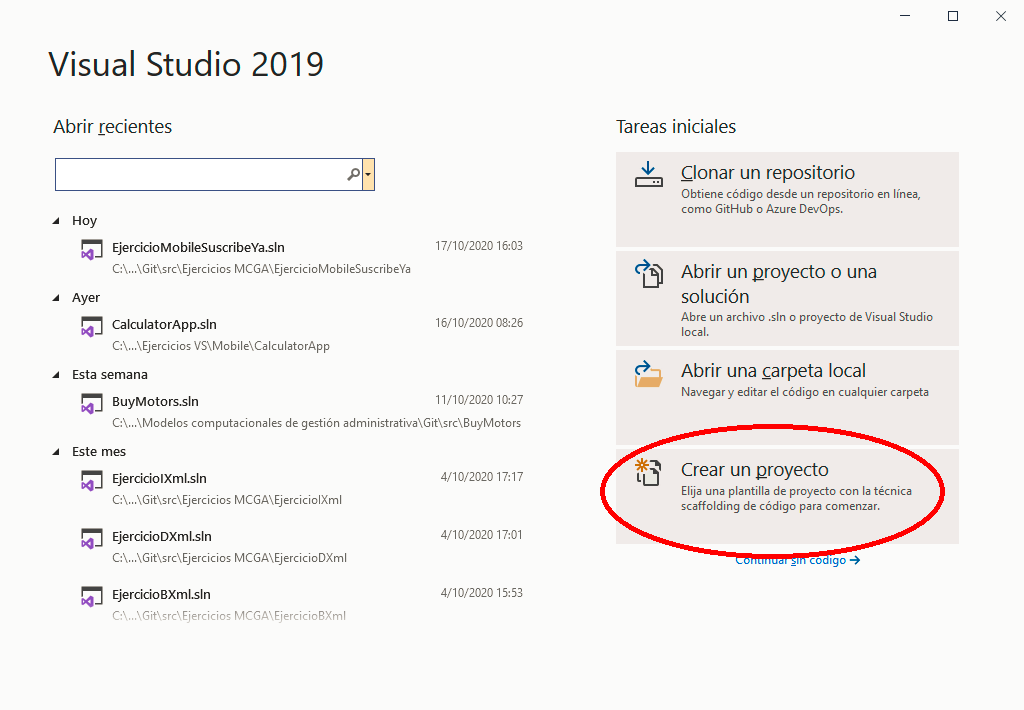
Es código cuya ejecución está administrada mediante un tiempo de ejecución (Common Language Runtime). CLR se encarga de tomar el código administrado, compilarlo en código máquina y, después, ejecutarlo.

Entre sus ventajas destacamos que el tiempo de ejecución proporciona varios servicios importantes, como la administración de memoria automática, los límites de seguridad, la seguridad de los tipos, etc. Además, es importante su posibilidad de ejecutarse en diferentes sistemas operativos ya que el código máquina no es creado hasta que debe utilizarse.

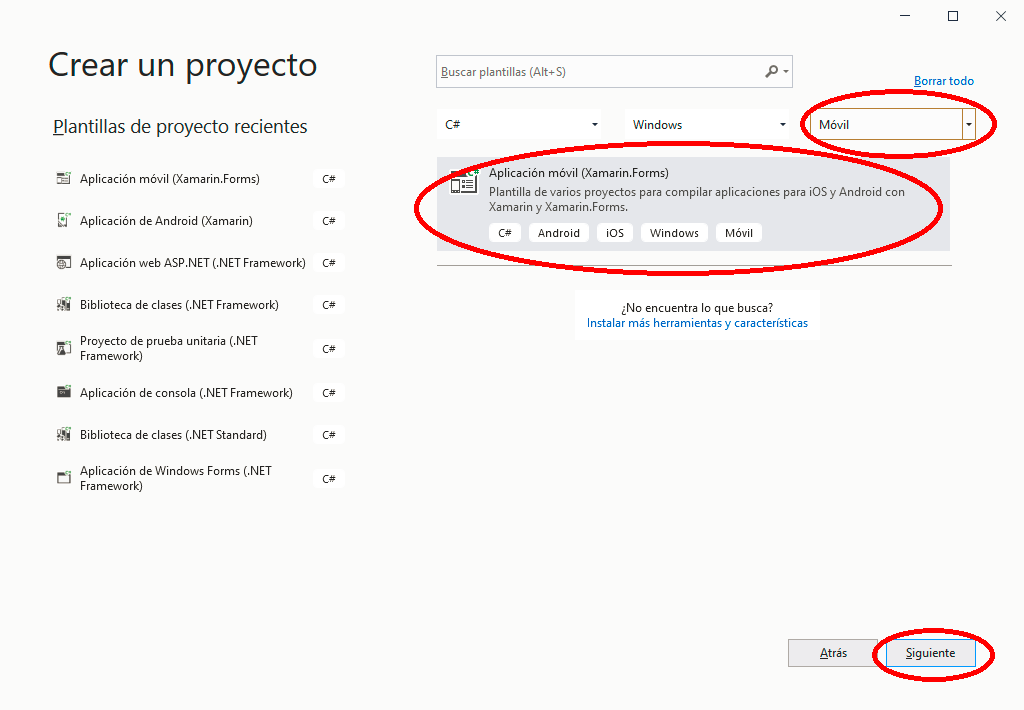
1. **Escriba la secuencia para crear una calculadora en algún emulador de mobile en Visual Studio 2017.**

Esta es la secuencia para crear una calculadora en el emulador de Android en Visual Studio 2019:

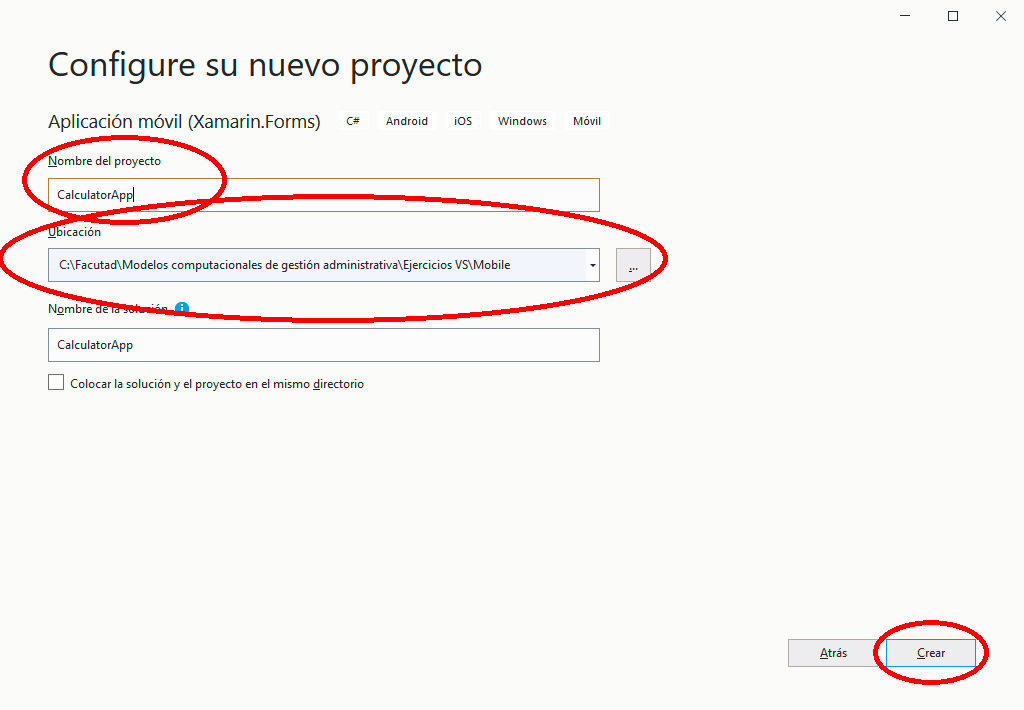
1. Se inicia Visual Studio 2019 y, en la ventana de inicio, se hace clic en *Crear un proyecto*:



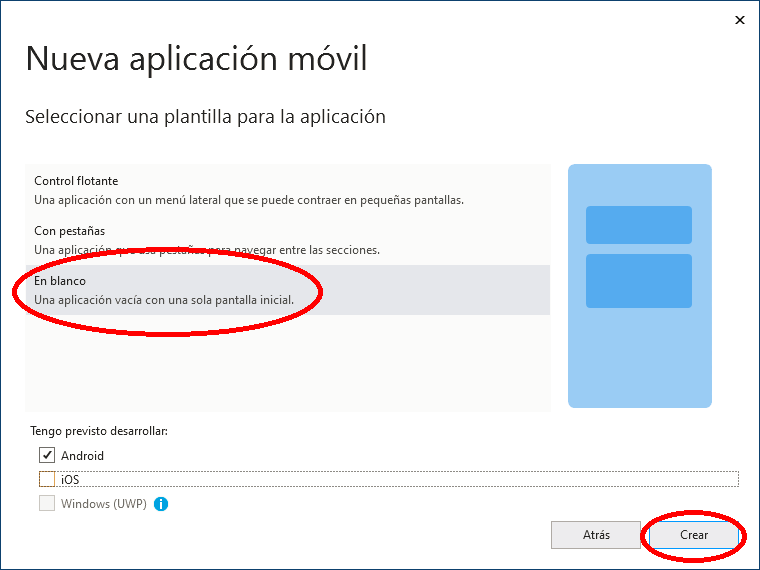
1. En la ventana *Crear un proyecto*, seleccionar *Móvil* en la lista desplegable *Tipo de proyecto*, elegir la plantilla *Aplicación móvil (Xamarin Forms)* y hacer clic en el botón *Siguiente*:



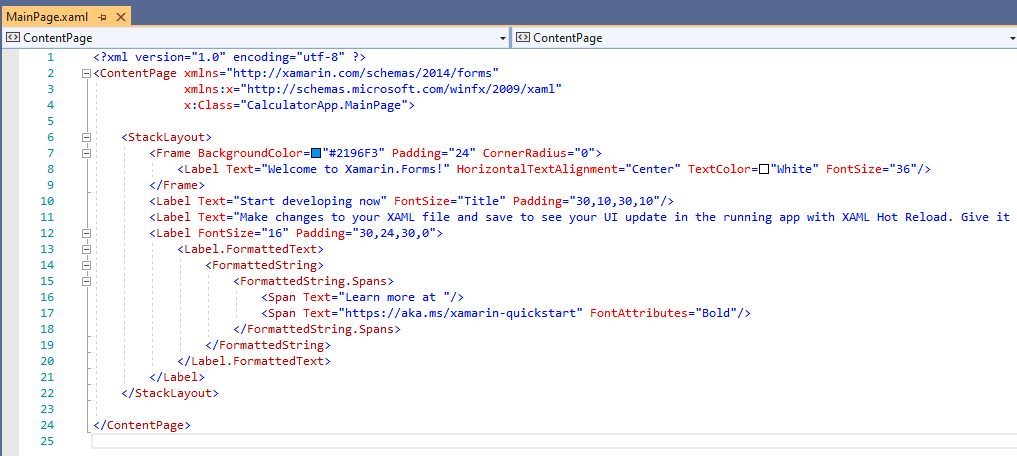
1. En el cuadro de diálogo *Configure su nuevo proyecto*, establecer *Nombre del proyecto* en *CalculatorApp*, elegir una ubicación adecuada para el proyecto y hacer clic en el botón *Crear*:



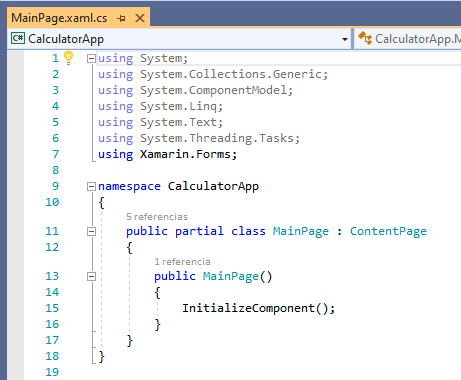
1. En el cuadro de diálogo *Nueva aplicación móvil*, hacer clic en *En blanco* y después en el botón *Crear*:



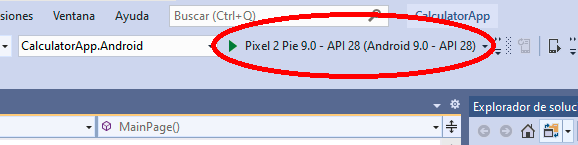
1. En el *Explorador de soluciones*, en el proyecto *CalculatorApp*, hacer doble clic en *MainPage.xaml* para abrirlo:



1. En *MainPage.xaml*, quitar todo el código de la plantilla y sustituirlo por el código correspondiente a la vista de la calculadora.
2. En el *Explorador de soluciones*, en el proyecto *CalculatorApp*, expandir *MainPage.xaml* y hacer doble clic en *MainPage.xaml.cs* para abrirlo:



1. En *MainPage.xaml.cs*, agregar el código necesario para manejar los controles que posee la calculadora y hacer el cálculo correspondiente, mostrando el resultado en algún control.
2. En la barra de herramientas de Visual Studio, pulsar el botón *Iniciar* (el botón triangular que parece un botón de reproducción) para iniciar la aplicación en la instancia elegida del emulador de Android:



1. Se iniciará el emulador de Android y se ejecutará la calculadora.
2. **¿Cuáles son los entornos de desarrollo más populares para crear dispositivos móviles?**

Los entornos de desarrollo más populares hoy en día para crear aplicaciones para dispositivos móviles (sin tener en cuenta los que se consideran nativos, como Android Studio por ejemplo) son los siguientes:

Native Script

NativeScript es un entorno de desarrollo de código abierto para crear aplicaciones móviles Android e iOS, con Angular, Vue.js, TypeScript o Javascript. Las principales ventajas de este entorno de programación es el rendimiento nativo. Solo habrá que definir las cosas una vez y NativeScript se adaptará para ejecutarse en todas partes, adaptando la UI de dispositivos y pantallas concretas.

Otro de sus puntos a favor es su escalabilidad. Se pueden acceder a las APIs de Android, pero se podrá reutilizar el SDK de Android, además de encontrar complementos gratuitos o plantillas en el marketplace de NativeScript.

Además, si se comienza a programar apps con NativeScript, resulta fácil de aprender. Pueden aplicarse conocimientos previos en Javascript y CSS, pero además existe una gran cantidad de recursos, tutoriales, ejemplos y una comunidad de usuarios fuerte.

React Native

React Native es un framework desarrollado por Facebook que permite crear aplicaciones para Android nativas utilizando Javascript. Esto es algo que ya permitían otros como Cordova, pero con la diferencia de que con React Native las apps se renderizan utilizando las vistas nativas, mientras que en Cordova se ejecutan dentro de una webview.

React Native tiene acceso a las API de Android, por lo que la experiencia de usuario y el rendimiento es similar al de una aplicación nativa. Otra de sus ventajas es la facilidad de ver los cambios con tan solo guardar. Gracias a Javascript, React Native permite iterar rápidamente, sin tener que esperar a que finalicen las compilaciones nativas.

Sin duda este framework es una apuesta segura, al contar con el respaldo de Facebook pero, sobre todo, con una gran comunidad de usuarios. En 2018, React Native llegó a tener el segundo mayor número de contribuciones en los repositorios de GitHub.

Ionic

Ionic es un framework de código abierto para desarrollar aplicaciones híbridas, en el que se combinan HTML5, CSS y Javascript. Gratis y de código abierto, Ionic ofrece una biblioteca de componentes, gestos y herramientas UI optimizados para dispositivos móviles con los que crear aplicaciones rápidas y altamente interactivas.

Ionic 5, versión más reciente del framework, está pensada para un rendimiento rápido en cualquier dispositivo. Con él se podrán crear apps Android rápidas, optimizadas y que integran las mejores prácticas recomendadas. Además, Ionic facilita acceder a las funcionalidades nativas a través de JavaScript. Está a disposición una librería de plugins nativos para acceder a la geolocalización, bluetooth o a la cámara sin grandes complejidades.

Xamarin

Xamarin es un entorno de desarrollo creado por Microsoft y que permite crear aplicaciones Android en C#. Es gratuito y va integrado en Visual Studio.

Pese a no estar desarrollando para Android en Java, Xamarin permite crear una aplicación nativa que asegura una buena experiencia de usuario, algo fundamental para cualquier app. Además, en caso de que se necesite desarrollar una aplicación multiplataforma, se podrá hacer con Xamarin.

Por otra parte, el lenguaje C# es más sencillo de aprender que otros lenguajes de programación. Así que si Java no es tu fuerte, puede que Xamarin sea la mejor solución para crear aplicaciones Android de calidad.

Kotlin

Kotlin es un lenguaje de programación de tipado estático que corre sobre la máquina virtual de Java. Sin embargo, crear aplicaciones Android con Kotlin tiene la ventaja de la simplicidad. En Kotlin se puede hacer más con menos código, lo que se traduce en una menor probabilidad de errores.

Otra de las ventajas de Kotlin es que es mucho más seguro que Java, y el código es más fácil de entender cuando lo lee otra persona. Además el compilador ayuda a detectar errores en la compilación, lo que contribuye a reducir posibles errores. Con Kotlin se podrá programar en Android, pero también para iOS puesto que es multiplataforma. Todo ello hace que Kotlin sea una alternativa segura y fiable frente a Java.

Flutter

Flutter es un SDK de Google pensado para crear aplicaciones Android e iOS, para móvil, web y escritorio, desde una única base de código. Su lenguaje de programación es Dart y se caracteriza porque hace posible un desarrollo rápido gracias a la funcionalidad Hot Reload. Con ella, pueden hacerse cambios sin tener que parar y arrancar la aplicación, con lo que la programación es más efectiva ya que se reducen las esperas.

Este framework permite crear funcionalidades rápidamente, con una experiencia de usuario nativa. Sin duda es uno de los SDK con más proyección, puesto que todo apunta a que será con el que se desarrollen los nuevos dispositivos Android.

1. **Encuentra alguna vinculación entre Azure y dispositivos móviles.**

Azure puede usarse como plataforma en la nube con la cual los dispositivos móviles se comunican para realizar ciertos tipos de operaciones. Una aplicación móvil puede comunicarse con una Web Api alojada en Azure para, por ejemplo, obtener datos que estén guardados en una base de datos remota (la cual también puede estar alojada en Azure). De esta forma, la aplicación móvil realiza una petición a dicha Web Api y ésta consulta en la base de datos correspondiente devolviendo los datos al dispositivo, realizando cierto procesamiento de esos datos en caso de ser necesario.

Un dispositivo móvil también puede recolectar datos y transferirlos a esa Web Api para realizar cierto procesamiento y devolver un resultado, o simplemente para almacenar los datos enviados en una base de datos.

Pero Azure no solamente ofrece un soporte de backend para aplicaciones productivas, también ofrece un servicio para probar y testear aplicaciones móviles que todavía están en etapa de desarrollo. El sistema permite la creación de entornos de ensayo privados para realizar pruebas antes de publicar o implementar una aplicación. De esta forma, se puede comprobar en tiempo real si el proyecto en el que se está trabajando cumple todas las expectativas cuando se traslada a un dispositivo móvil.

Mediante los servicios móviles de Azure, se pueden realizar distintos proyectos relacionados con aplicaciones para dispositivos móviles. Esto se consigue con una plataforma de desarrollo escalable, diseñada para desarrolladores e integradores de aplicaciones y sistemas corporativos. Este entorno ofrece funcionalidades como:

* Desarrollo de aplicaciones nativas y multiplataforma: los servicios móviles de Azure dan la posibilidad de usar kits de desarrollo de software (SDK) nativos en la nube. Estos SDK en Azure permiten crear aplicaciones nativas (iOS, Android, Windows Phone) y multiplataforma (Xamarin o PhoneGap).
* Inicio de sesión en sistemas corporativos: si necesita acceder a sus recursos empresariales desde cualquier lugar, puede hacerlo con una aplicación en la nube de Azure. Además, puede añadir ciertos protocolos para autentificación en dos pasos y aumentar la seguridad de acceso a sus sistemas corporativos.
* Gestión de datos de aplicaciones y envío de notificaciones push: Microsoft Azure permitealmacenar los datos de miles de usuarios de su aplicación y enviarles instantáneamente notificaciones push (peticiones desde servidor a cliente). Estas notificaciones pueden programarse para enviarse con información personalizada en el momento más adecuado.
* Sincronización de aplicaciones tras uso sin conexión: Azure permite que una aplicación utilizada sin conexión a internet sincronice datos en segundo plano al volver a tener acceso a la red. Esto se utiliza cuando los empleados tengan que utilizar las aplicaciones sin conectividad para que los datos puedan sincronizarse con el sistema central.
* Conexión con datos locales: las aplicaciones móviles creadas en Azure permiten utilizar los datos de su *datacenter* local. Mediante redes híbridas y sistemas VPN podrá acceder de forma segura a la información interna de su empresa con el móvil.