

## Tecnologías para el desarrollo

<b>Tecnologías para el desarrollo</b>	<b>1</b>
¿Qué aprenderás?	2
Introducción	2
Ecosistemas de soluciones	3
.NET	3
Plataforma Java	4
Swift	5
Ruby	7
Ruby on Rails	8
Python	9
Django	9
Ionic	10
Xamarin	11
Kotlin	12



**¡Comencemos!**

## ¿Qué aprenderás?

- Distinguir tecnologías para el desarrollo de aplicaciones web y móviles para la construcción de prototipos digitales.
- Identificar la importancia de Rails para la construcción de prototipos digitales.

## Introducción

Los emprendedores dedicados a las startups digitales tienen la obligación de tener ciertos conocimientos de tecnología, programación, saber cómo funciona una aplicación, conocer los frameworks disponibles, las tendencias de la industria, entre otras cosas.

Si eres tú quien desarrolla la solución está claro el porqué, pero si no, tendrás que relacionarte con tu socio/a TI o el equipo externo a cargo de esta labor. Te agrega mucho valor profesional el poder conversar con desarrolladores utilizando los conceptos correctos.

Si ya tenemos medianamente claro el problema, nos enamoramos de él, conocemos a los clientes y usuarios y hemos validado un prototipo de baja resolución, entonces nos tocará tomar decisiones sobre lo técnico: **¿Qué tecnología es la que más se acomoda al problema que quiero resolver?**

En este apartado entregaremos orientaciones para apoyarte en esa decisión.

**¡Vamos con todo!**



## Ecosistemas de soluciones

Existen muchas soluciones disponibles en el mercado y cada vez surgen nuevas dada la constante evolución tecnológica. Presentaremos a continuación las más comúnmente utilizadas y sus principales características.

### .NET



Imagen 1. Logo Microsoft .NET Framework

Fuente: [mistertek.com](http://mistertek.com)

Es una suite de herramientas creada por la empresa Microsoft que simplifica la construcción de aplicaciones. El objetivo de .NET es poder desarrollar aplicaciones que fueran independientes de la arquitectura física (procesador) y del sistema operativo. Para esto se creó un protocolo de intercambio de información llamado XML, muy similar en sintaxis a HTML al ser ambos lenguajes de marcado (por etiquetas).

Es frecuente escuchar que .NET es competencia directa de JAVA o algunos frameworks basados en PHP desde el punto de vista de la propuesta de valor: poder desarrollar sin pensar en sistemas operativos y arquitecturas donde se ejecute la aplicación.

Cabe destacar que .NET es multipropósito y multiplataforma: se puede usar tanto para desarrollo web como para aplicaciones de escritorio o dispositivos móviles con sistema operativo iOS, Android o Windows Phone.



Cuando nos dicen que una aplicación está desarrollada en .NET se está cometiendo un error, ya que se usa como suite pero se desarrolla en alguno de los lenguajes soportados como son C#, C++, Visual Basic, .NET o F#.

## Plataforma Java



Imagen 2. Logo Java  
Fuente: [Lorenaspy Site](#)

Es un entorno de desarrollo y ejecución de aplicaciones escritas en diferentes lenguajes, siendo el principal Java. El Entorno Java funciona como una máquina virtual que se monta sobre el sistema operativo para ejecutar aplicaciones de forma “segura”.

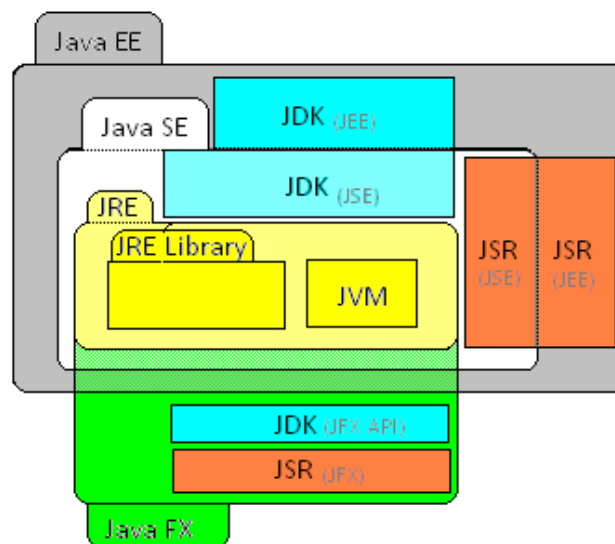


Imagen 3. Plataforma Java  
Fuente: [Wikipedia](#)

El Entorno Java fue creado y comercializado inicialmente por **Sun Microsystems**. En la actualidad depende de Oracle Corporation. Hay más de 3 mil millones de dispositivos ejecutando Java, 12 mil millones de desarrolladores en el mundo.

El lenguaje Java fue creado por el canadiense James Gosling en 1995, es un lenguaje de alto nivel que cuenta con librerías para manipular hardware, utiliza máquinas virtuales que permiten correr el código en cualquier momento y en cualquier sistema operativo que tenga instalado el entorno. Está basado en C++ y está orientado a objetos, multihilo y tiene características de programación funcional.

Sin duda Java es uno de los grandes lenguajes a nivel mundial, siendo utilizado por grandes compañías como Google, Amazon, Über y bancos que prefieren optar por la seguridad del respaldo que tienen con Oracle, lo portable y mantenible. Por la misma razón es muy raro encontrar una startup que elija este entorno para desarrollar sus soluciones inicialmente.

Te dejamos un ejemplo de un “Hola Mundo” escrito en Java:

```
public class HolaMundo {  
    public static void main(String[] args) {  
        System.out.println("Hola Mundo");  
    }  
}
```

## Swift

Es un lenguaje multipropósito creado recientemente por la empresa Apple en 2014. Es multiparadigma. Orientado a objetos, protocolos, funcional, programación imperativa). Está enfocado principalmente en el desarrollo de aplicaciones en el ecosistema de Apple: iOS, iPadOS, watchOS, tvOS. Puede ser instalado en OSX o Linux. El gran beneficio es que desarrollas una vez tu aplicación y luego solo ajustas las versiones para los dispositivos de una forma muy simple con Xcode.

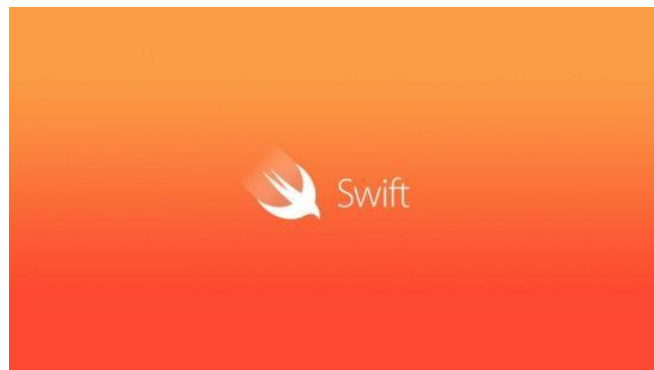


Imagen 4. Logo Swift  
Fuente: [nobbot.com](http://nobbot.com)

Se integra nativamente a los frameworks Cocoa y Cocoa Touch, puede usar cualquier biblioteca de Objective-C. Tiene un sintaxis muy simple y hace recordar a lenguajes de programación modernos como Python o Ruby.

Un ejemplo de un “Hola Mundo” que deja en evidencia lo simple de Swift :

```
print("Hola mundo!")
```

Si bien esta es una opción para usuarios del ecosistema Apple (iOS), no debemos perder de vista que Android es un actor muy relevante en el mercado.

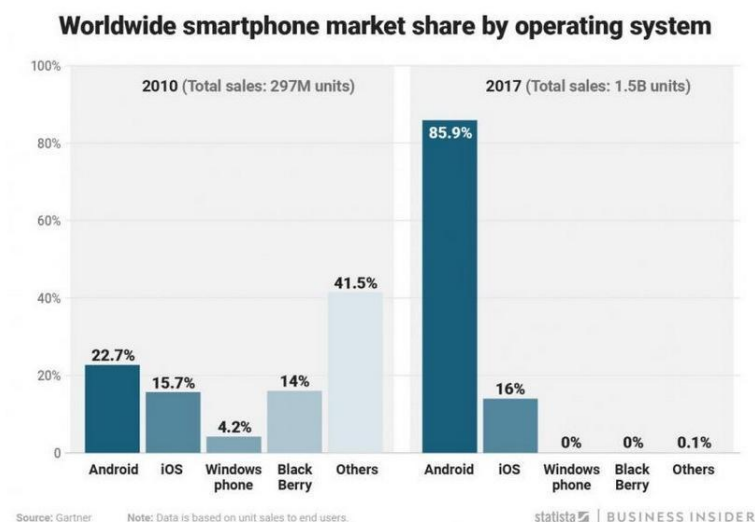


Imagen 5. Participación de Mercado por Sistema Operativo

Fuente: [computerhoy.com](http://computerhoy.com)

Cabe mencionar que, aún cuando sólo el 16% de los móviles a nivel mundial usan iOS y el 3% de los computadores usan OS X, este grupo de mercado es el que tiene las más altas tasas de compra de aplicaciones y suscripciones.

Otro beneficio de los usuarios de iPhone o iPads es que el 90% usa las últimas dos versiones del sistema operativo. Esto significa que puedes crear aplicaciones pensando en muy pocos casos de versiones de sistema operativo y tamaños de pantallas.

## Ruby

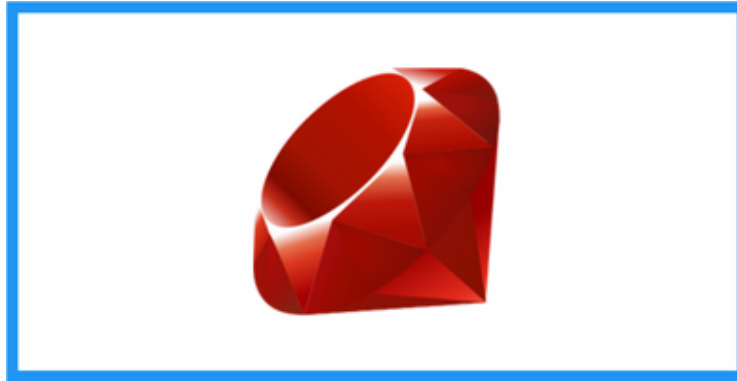


Imagen 6. Logo Ruby

Fuente: [lenguajesdeprogramacion.net](http://lenguajesdeprogramacion.net)

Es un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad. Su elegante sintaxis se siente natural al leerla y fácil al escribirla (en inglés). Por esto último se suele decir que Ruby está pensada para la felicidad de los programadores.

Fue creado en 1995 por Yukihiro “Matz” Matsumoto, quien juntó características de sus lenguajes favoritos para crear un lenguaje orientado a objetos, programación funcional y la imperativa. En palabras de Matz:



“Ruby es simple en apariencia, pero complejo por dentro, como el cuerpo humano.”

Un “Hello World” en Ruby es tan simple como en Python y Swift

```
puts "Hello World!"
```

A diferencia de otros lenguajes de programación donde los objetos primitivos como números no son objetos, en Ruby todo es un objeto. Se les puede asignar propiedades y acciones a toda información y código.

Ruby es flexible desde el punto de vista que permite a cualquier usuario modificarlo libremente. El código de Ruby puede ser redefinido como quieras.

Ruby es fácilmente portable: se desarrolla mayoritariamente en GNU/Linux, pero corre en varios tipos de UNIX, macOS, Windows, DOS, BeOS, OS/2, etc.

## Ruby on Rails



Imagen 7. Logo Ruby on Rails.

Fuente: [programacion.net](http://programacion.net)

También conocido como RoR, es un framework (entorno de trabajo) de código abierto basado en Ruby para el desarrollo de aplicaciones web. Sigue un paradigma llamado MVC (Modelo Vista Controlador) y permite crear aplicaciones web con muy poco código de por medio. RoR se distribuye como una gema o RubyGem, que es el canal oficial de distribución de paquetes, bibliotecas y aplicaciones de Ruby. El sitio oficial del lenguaje en la web es [rubyonrails.org](http://rubyonrails.org).

Fue creado en 2005 por David Heinemeier Hansson, (@dhh en Twitter).

Este framework sigue la filosofía DRY: Don't Repeat Yourself que significa "No te repitas", haciendo referencia a que las definiciones o métodos deberían hacerse sólo una vez.

RoR es ideal para prototipar ideas rápidamente, hasta el desarrollo de complejas aplicaciones con cientos de miles de usuarios. Dentro de las aplicaciones más famosas hechas con Ruby on Rails (o que lo fueron en algún momento) está Basecamp, GitHub, Shopify, Airbnb, Twitch, SoundCloud, Hulu, Zendesk, Square y Cookpad entre otras.

Optimizado por la felicidad del programador/a, RoR es una de las opciones ideales para personas que nunca han programado y quieren aprender y desarrollar sus propias soluciones desde cero.

Comenzar una aplicación es tan simple como esta línea de código:

```
rails new my_app_name
```

Existen otros frameworks para el desarrollo de aplicaciones web basados en Ruby como Sinatra.



## Python



Imagen 8. Logo Python

Fuente: [python.org](https://www.python.org)

El Lenguaje Python fue inventado alrededor de 1990 por el científico en computación holandés Guido van Rossem y su nombre es un tributo al grupo cómico Monty Python del cual Guido es admirador. El sitio oficial del lenguaje en la web es <https://www.python.org>.

Una de las mayores fortalezas de Python es su comunidad de desarrolladores, liderada por la PSF (Python Software Foundation) que le mantienen y dan soporte. Es usado con frecuencia por la comunidad científica, en especial por la NASA quienes aprovechan la sintaxis clara, su avanzada integración con procedimientos matemáticos complejos y la ejecución de logaritmos científicos que, a pesar de ser un lenguaje interpretado, hay forma de compilarlo para emular el rendimiento de otros lenguajes más rápidos. Esto lo hace un lenguaje de preferencia en el mundo científico.

## Django

Es el framework para desarrollo de aplicaciones web basado en Python más famoso usado por los desarrolladores. Django usa el patrón de diseño Modelo Vista Controlador (MVC). El objetivo de Django es facilitar la creación de sitios web dinámicos y complejos mediante el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio DRY (Don't Repeat Yourself).



Imagen 9: Logo Django.

Fuente: [edgica.com](http://edgica.com)

Una característica diferenciadora de Django es que proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como parte de cualquier página hecha con Django y que puede administrar varias páginas a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios (incluyendo una asignación detallada de permisos). Esta última característica no viene por defecto en Ruby on Rails y se debe implementar como una nueva gema del proyecto.

## Ionic

Es un framework reciente que está adquiriendo popularidad. Es una herramienta gratuita para desarrollar apps basadas en HTML5, CSS y JavaScript. Es libre y de código abierto, es decir, open source. Trabaja con AngularJS, lo que entrega una arquitectura robusta para el desarrollo de apps.

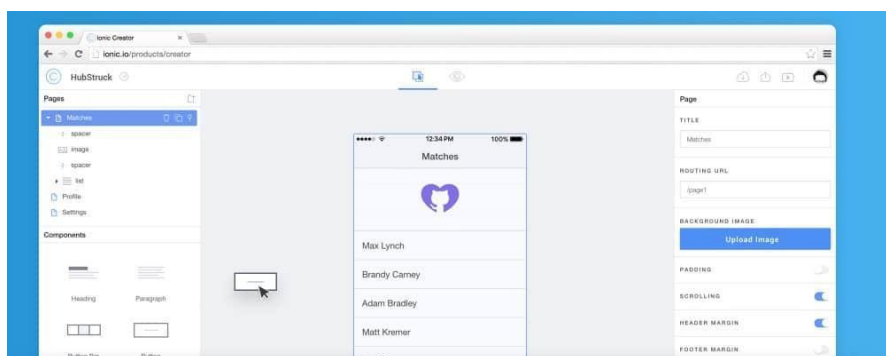


Imagen 10. Entorno Ionic.

Fuente: [randyvarela.es](http://randyvarela.es)

Al ser un lenguaje híbrido, su rendimiento nunca podrá ser como el de una aplicación nativa, y no es recomendable para proyectos de gran envergadura. También posee algunas restricciones para los componentes iOS.

## Xamarin

Xamarin es utilizado para el desarrollo de aplicaciones móviles multiplataforma y usa el lenguaje, C #, para crear aplicaciones para todas las plataformas móviles. Se compila de forma nativa, lo que lo convierte en una opción de acceso para crear aplicaciones de alto rendimiento con un aspecto nativo.

Es un lenguaje con elementos de seguridad que evita que el código tenga un comportamiento inesperado.

Pese a estas ventajas, también posee complicaciones como: acceso limitado a bibliotecas, alto costo para uso personal y profesional, comunidad de desarrolladores pequeña, entre otros.

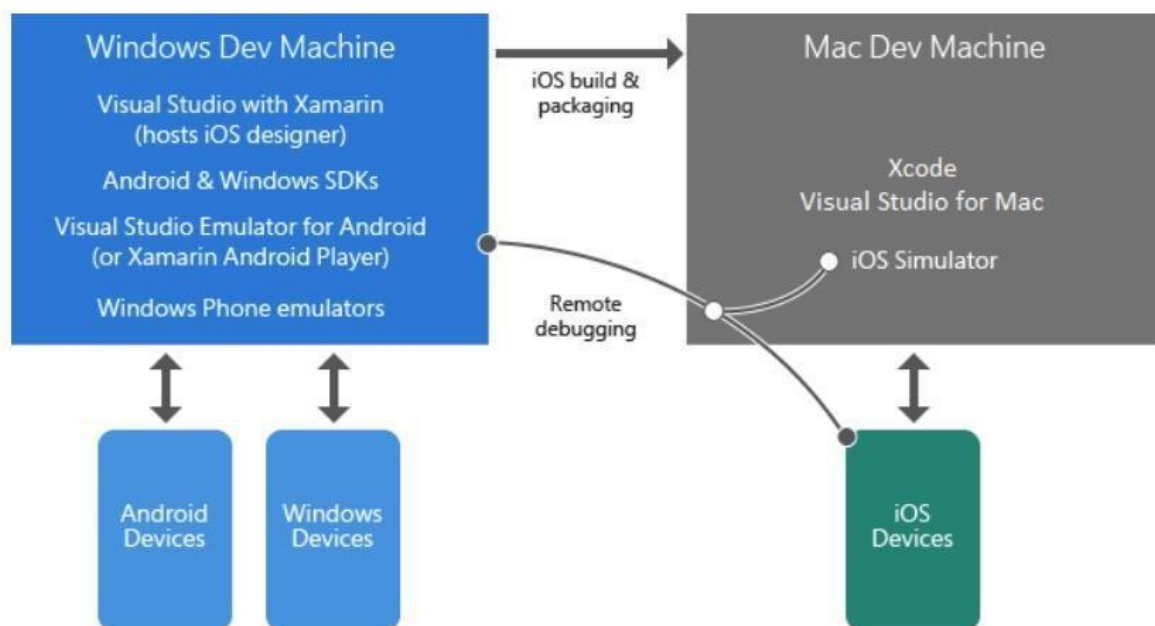


Imagen 11. Estructura Proyecto Xamarin

Fuente: [Blog de Inmediatum](#)

## Kotlin

Es un lenguaje de programación que es utilizado como lenguaje Android, dirigido a desarrollos back end. Se basa en Java, utilizando sus mejores elementos.

Dentro de sus principales características podemos considerar que es 100% compatible con librerías Java, está orientado a objetos con un estilo funcional y tiene una codificación simplificada.

Algunas dificultades que se pueden presentar es la comprensión inicial del código, tiene una comunidad de soporte pequeña al ser un lenguaje nuevo y los tiempos de ejecución pueden ser un poco más extensos.



Imagen 12. Representación Kotlin.

Fuente: [genbeta.com](http://genbeta.com)



### ¡Importante!

Elegir un lenguaje para nuestra solución puede tener dependencia de múltiples factores, entre ellos podemos mencionar:

- Facilidad de Uso.
- Soporte del fabricante.
- Comunidad de desarrollo.
- Experiencia en el equipo de trabajo.
- Necesidad de considerar determinados dispositivos y utilizar sus recursos.