



HEINSOHN
BUSINESS TECHNOLOGY

Introducción Microsoft .NET



Microsoft Partner

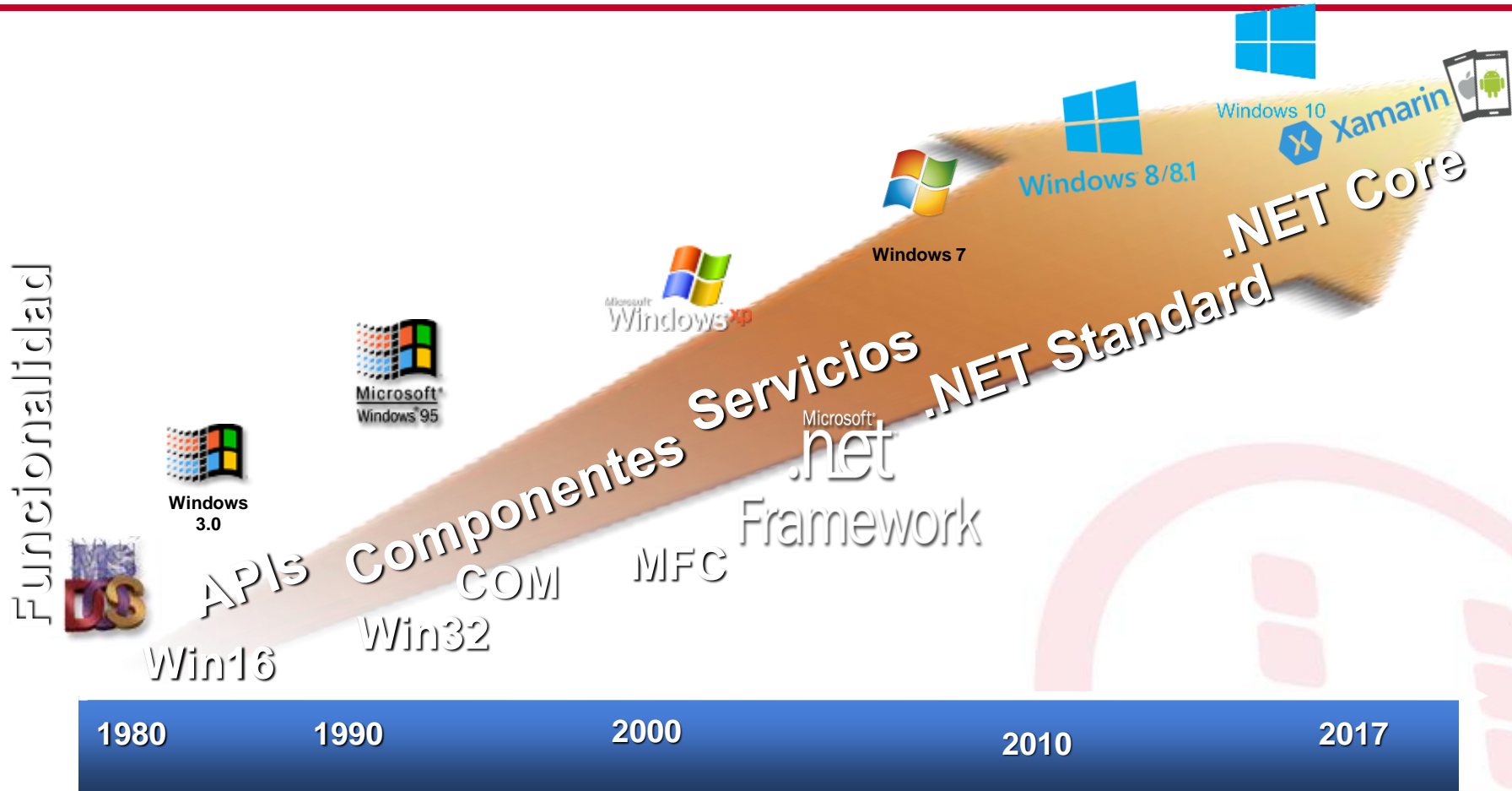
Gold Software Development

Gold Web Development

Gold Software Asset Management

Presentar una introducción a la plataforma de desarrollo Microsoft .NET, describiendo sus principios básicos de funcionamiento, su arquitectura de componentes y sus principales bibliotecas reutilizables.

- Introducción a Microsoft .NET
- Componentes Fundamentales
- Funcionamiento Interno del CLR
- Bibliotecas Principales
- Ventajas de .NET
- Herramientas de Desarrollo .NET



- .NET no es un Sistema Operativo
- .NET no es un Lenguaje de Programación
- .NET no es un Entorno de Desarrollo
- .NET no es un Servidor de Aplicaciones
- .NET no es un producto empaquetado que se pueda comprar como tal

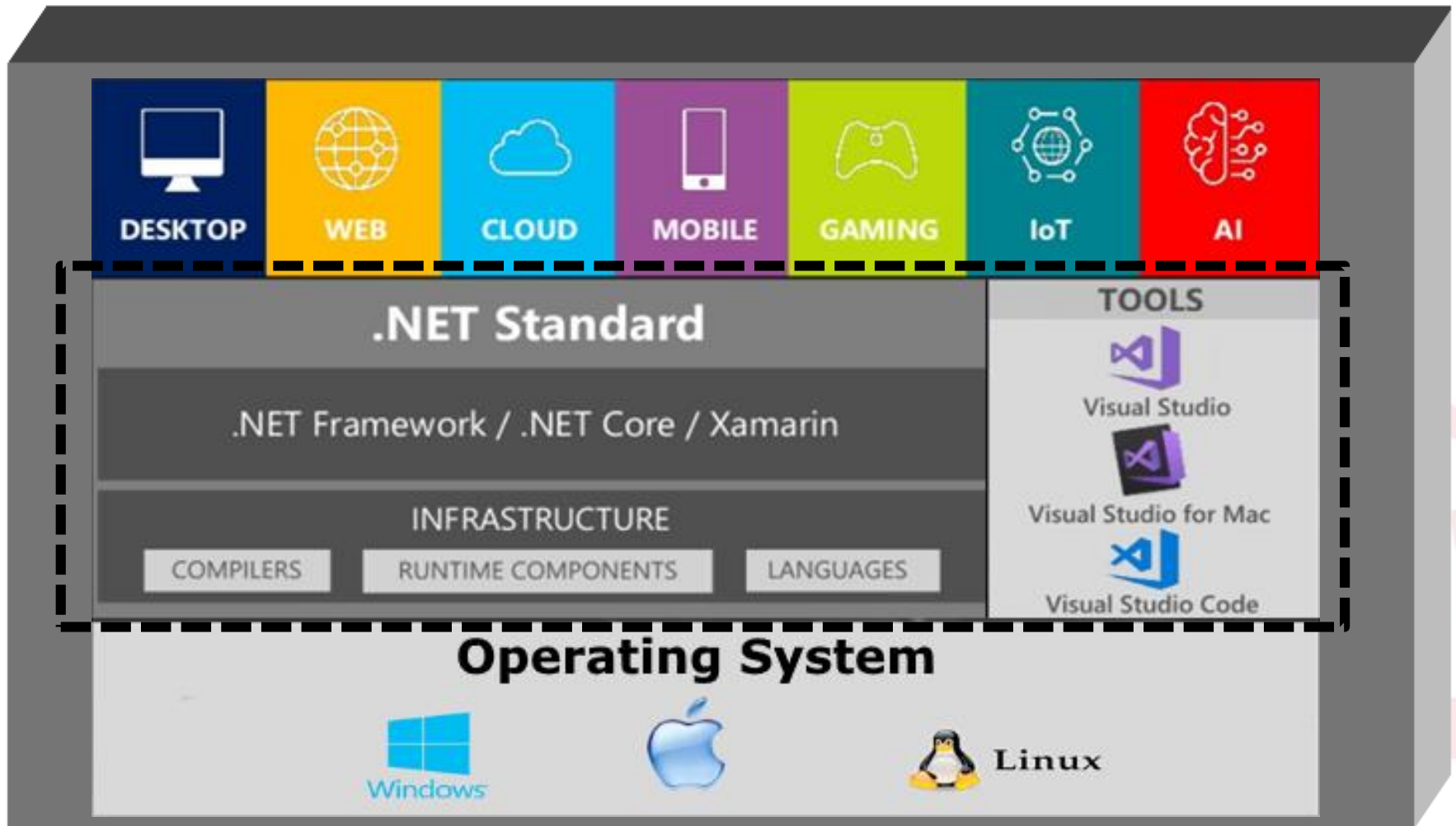
Es una plataforma de Desarrollo compuesta de:

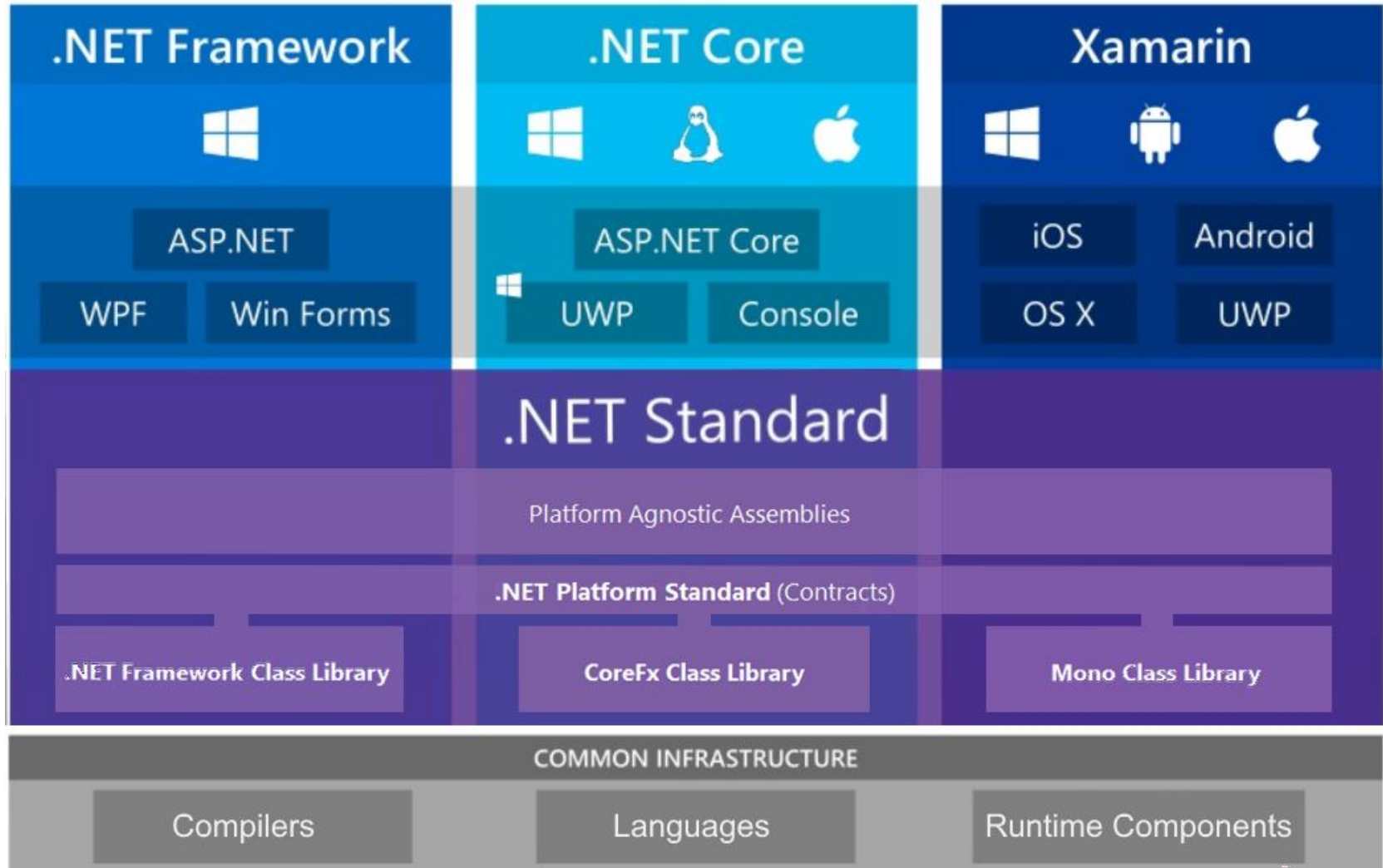
- Entorno de Ejecución (Runtime)
- Bibliotecas de Funcionalidad (Class Library)
- Lenguajes de Programación
- Compiladores
- Herramientas de Desarrollo (IDE & Tools)
- Guías de Arquitectura

Es la evolución de la plataforma COM

- Plataforma de ejecución intermedia
- 100% Orientada a Objetos
- Multilenguaje
- Plataforma Empresarial de Misión Crítica
- Modelo de Programación único para todo tipo de aplicaciones y dispositivos de hardware
- Se integra fácilmente con aplicaciones existentes desarrolladas en plataformas Microsoft
- Se integra fácilmente con aplicaciones desarrolladas en otras plataformas

Plataforma de Ejecución Intermedia





- Paquete de software fundamental de la plataforma .NET. Incluye:
 - Entorno de Ejecución (Runtime)
 - Bibliotecas de Funcionalidad (Class Library)
- Se distribuye en forma libre y gratuita
- Existen tres variantes principales:
 - .NET Framework Redistributable Package
 - .NET Framework SDK
 - .NET Compact Framework

¿Dónde instalar el .NET Framework?

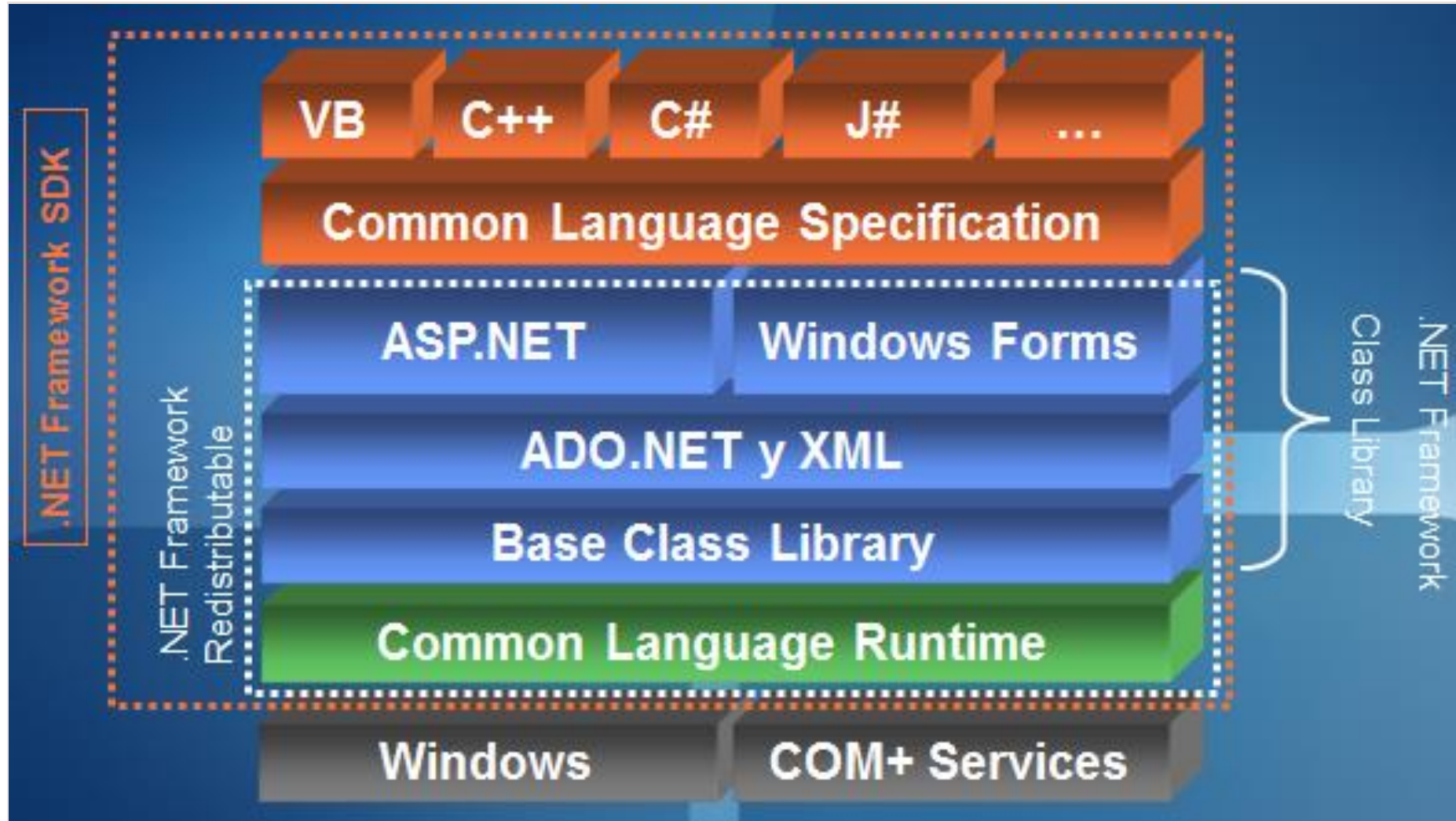
	Cliente	Servidor
Aplicación de Escritorio	✓	✓ *
Aplicación Web		✓
Aplicación de Consola	✓	✓ *

*** Sólo si la aplicación es distribuida**

- .Net Core es una plataforma de desarrollo que tiene un uso general y es de código abierto.
- Es multiplataforma, admite macOS, Windows y Linux.
- Se utiliza para compilar aplicaciones de nube o IoT.
- .NET Core es compatible con .NET Framework, Xamarin y Mono utilizando .NET Standard
- Incluye herramientas de líneas de comando que se utilizan para el desarrollo local y en escenarios de integración continua.
- Los lenguaje VB, C# y F# pueden usarse para escribir aplicaciones para .NET Core

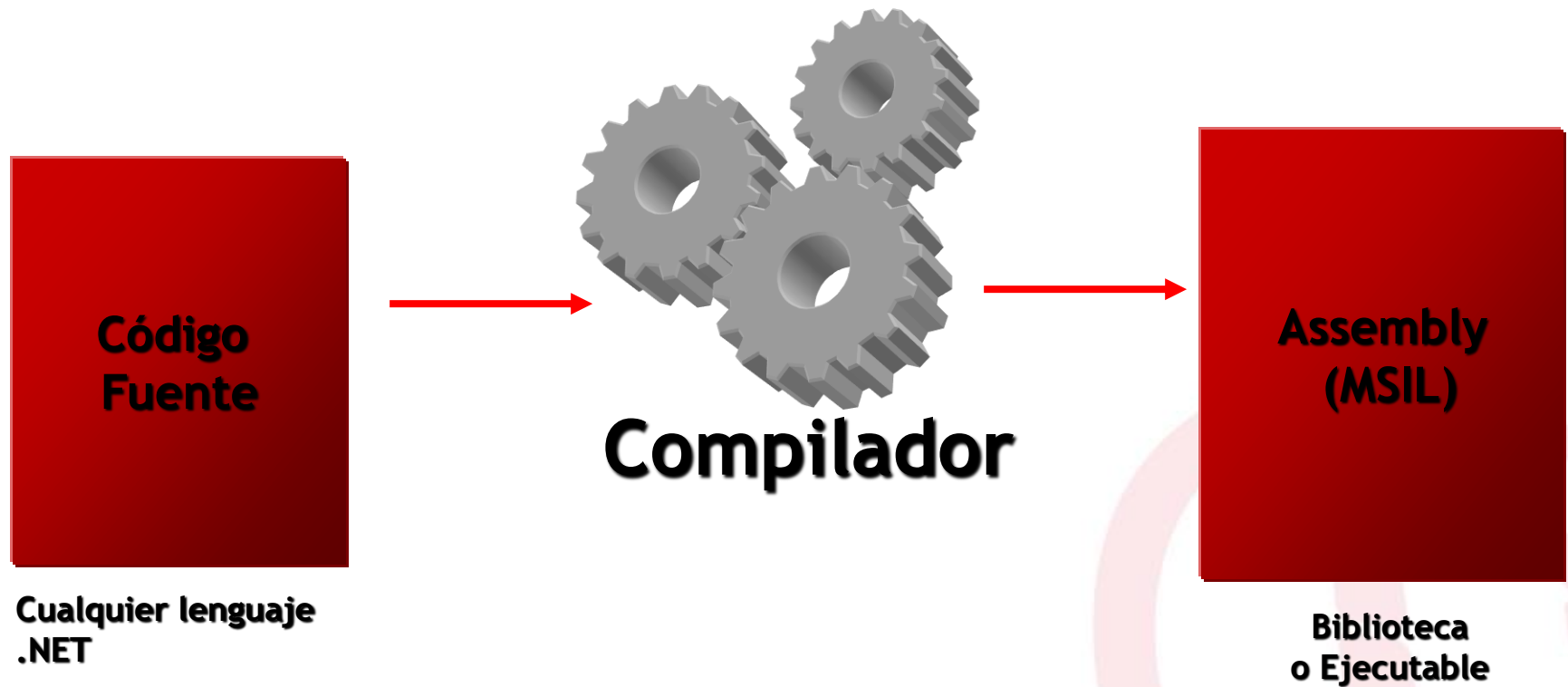
- Introducción a Microsoft .NET
- Componentes Fundamentales
- Funcionamiento Interno del CLR
- Bibliotecas Principales
- Ventajas de .NET
- Herramientas de Desarrollo .NET

Arquitectura del .NET Framework



- El CLR es el motor de ejecución (runtime) de .NET
- Características
 - Gestión automática de memoria (Garbage Collector)
 - Gestión de errores consistente (Excepciones)
 - Ejecución basada en componentes (Assemblies)
 - Gestión de Seguridad
 - Multithreading

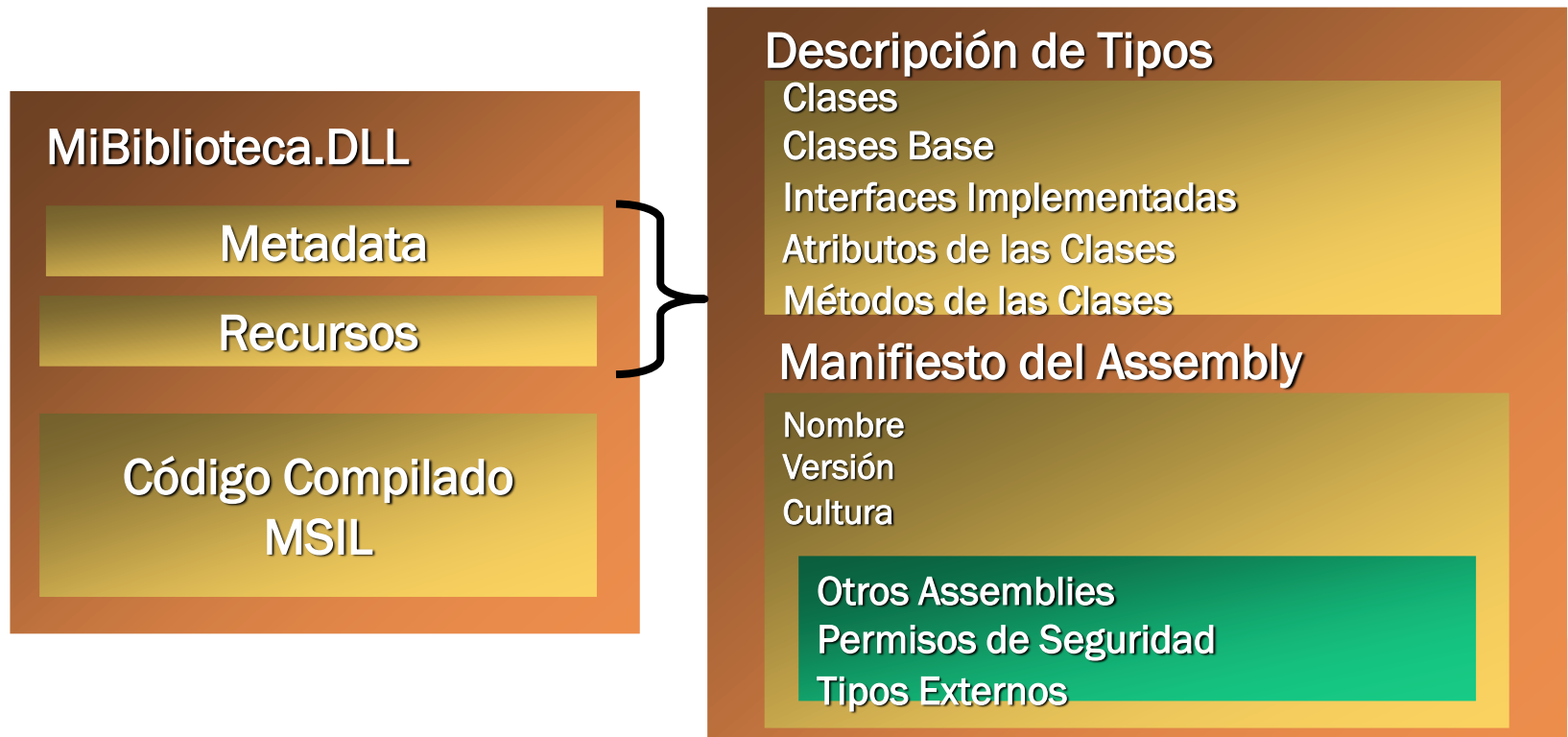
CLR - Proceso de Compilación




```
.method private hidebysig static void Main(string[] args) cil  
    managed {  
.entrypoint  
maxstack 8  
L_0000: ldstr "Hola Mundo"  
L_0005: call void [mscorlib]System.Console::WriteLine(string)  
L_000a: ret  
}
```

¿Qué es un “Assembly”?

- Un Assembly es la unidad mínima de ejecución, distribución, instalación y versionado de aplicaciones .NET



.NET Framework Class Library

System.Web

Services

Description

Discovery

Protocols

Caching

Configuration

UI

HtmlControls

WebControls

Security

SessionState

System.Windows.Forms

Design

ComponentModel

System.Drawing

Drawing2D

Imaging

Printing

Text

System.Data

OleDb

Common

Odbc

SqlClient

System.Xml

XSLT

XPath

Serialization

System

Collections

Configuration

Diagnostics

Globalization

IO

Net

Reflection

Resources

Security

ServiceProcess

Text

Threading

Runtime

InteropServices

Remoting

Serialization

- .NET posee un único runtime (el CLR) y un único conjunto de bibliotecas para todos los lenguajes
- No hay diferencias notorias de performance entre los lenguajes provistos por Microsoft
- El lenguaje a utilizar, en gral., dependerá de su experiencia previa con otros lenguajes o de gustos personales
 - Si conoce Java, Delphi, C++, etc. C#
 - Si conoce Visual Basic o VBScript, VB.NET
- Los tipos de aplicaciones .NET son INDEPENDIENTES del lenguaje que elija

- Introducción a Microsoft .NET
- Componentes Fundamentales
- Funcionamiento Interno del CLR
- Bibliotecas Principales
- Ventajas de .NET
- Herramientas de Desarrollo .NET

Modelo de Ejecución del CLR

Desarrollo

Assembly

Instalación

Instalación

Assembly en
la máquina
destino

Policy

```
class System.Security.Policy.PolicyStatement {  
    public PolicyStatement(PolicyStatementKind kind, string name, string permissionSetName)  
    {  
        this.kind = kind;  
        this.name = name;  
        this.permissionSetName = permissionSetName;  
    }  
    public PolicyStatementKind kind;  
    public string name;  
    public string permissionSetName;  
    public bool isPublicKeyTokenRequired;  
    public bool isNameRequired;  
    public bool isPermissionSetNameRequired;  
    public bool isPublicKeyTokenRequired;  
    public bool isNameRequired;  
    public bool isPermissionSetNameRequired;  
    public bool isPublicKeyTokenRequired;  
    public bool isNameRequired;  
    public bool isPermissionSetNameRequired;  
}
```

Ejecución

Compilador
JIT

Class
Loader

Seguridad

Assembly
Loader

.exe Nativo
+
Tabla GC

Code
Manager

Debug Engine

Garbage Collector

Manejador Excepciones

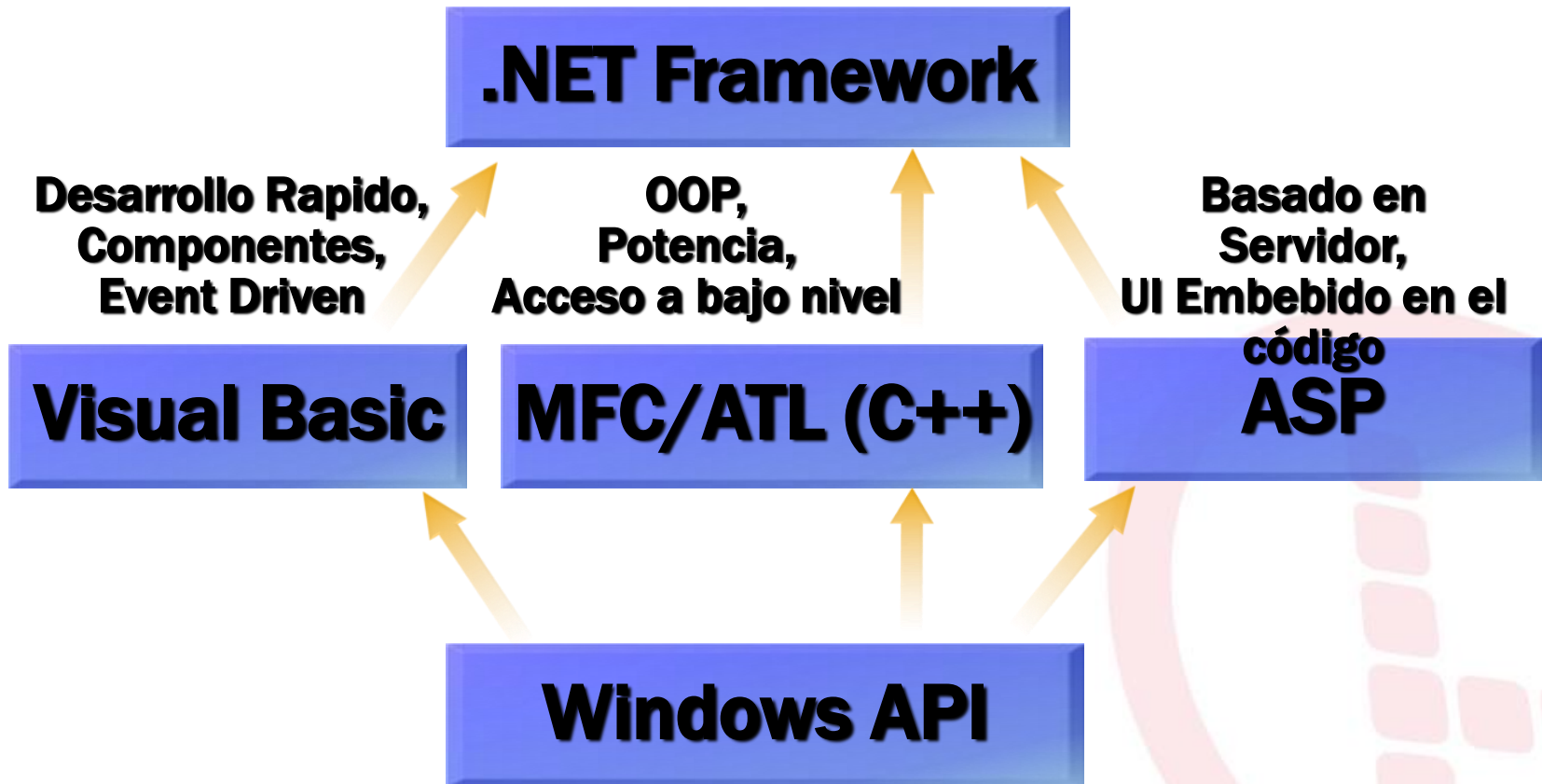
Soporte MultiThread

COM Interop

- Introducción a Microsoft .NET
- Componentes Fundamentales
- Funcionamiento Interno del CLR
- Bibliotecas Principales
- **Ventajas de .NET**
- **Herramientas de Desarrollo .NET**

- Unifica los modelos de programación
- Simplifica aún más el desarrollo
- Provee un Entorno de Ejecución robusto y seguro
- Es independiente del lenguaje de programación
- Interoperabilidad con código existente
- Simplifica la instalación y administración de las aplicaciones

**API consistente mas allá del lenguaje
o del modelo de programación**

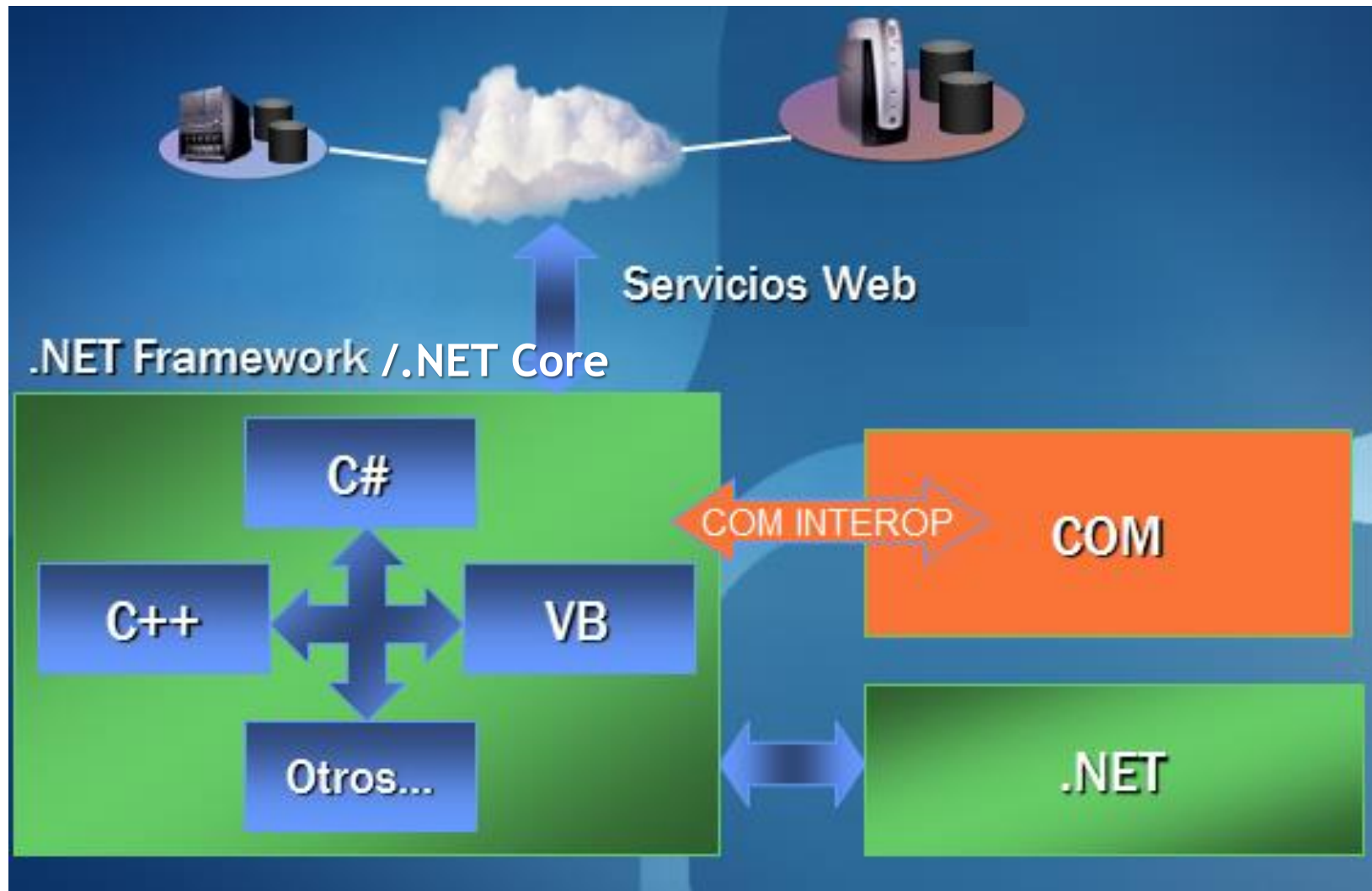


- Alto nivel de abstracción
 - No mas accesos COM a bajo nivel
 - Orientado a Objetos desde el Núcleo
- Sistema de tipos unificado (CTS)
 - Todo es un objeto, no mas variants
- Componentes de Software
 - Propiedades, métodos, eventos, y atributos incluidos en la construcción de clases
- API organizada en forma Jerárquica

- Gestión automática de la memoria
 - Todos los objetos son administrados por el Garbage Collector
- Manejo de Excepciones
- Fuertemente tipado
 - Solo casteos seguros
 - Inicialización de variables obligatoria
- Instalación con Cero Impacto
 - No requiere registración en la Registry

- Libertad en la elección del lenguaje
 - Todas las facilidades de la plataforma .NET están disponibles a todos los lenguajes de programación .NET
 - Los componentes de una aplicación .NET pueden ser escritos en distintos lenguajes de alto nivel compatibles con la plataforma
- Herramientas compartidas
 - Debuggers, profilers, analizadores de código, y otras trabajan para todos los lenguajes

- Unidades de Ensamblado (“Assemblies”)
 - Mínima unidad de distribución, versionado y administración de seguridad de aplicaciones .NET
 - Auto-descriptas a través de un manifiesto (“manifest”)
- Instalaciones Cero-impacto
 - Aplicaciones y componentes pueden ser compartidas o privadas
- Versioning
 - Múltiples versiones del mismo componente pueden co-existir, aún en el mismo proceso



- Introducción a Microsoft .NET
- Componentes Fundamentales
- Funcionamiento Interno del CLR
- Bibliotecas Principales
- Ventajas de .NET
- **Herramientas de Desarrollo .NET**

Visual Studio



Herramientas de
desarrollo



Servicios Cloud



Software



Formación y Soporte

<https://www.visualstudio.com/es>

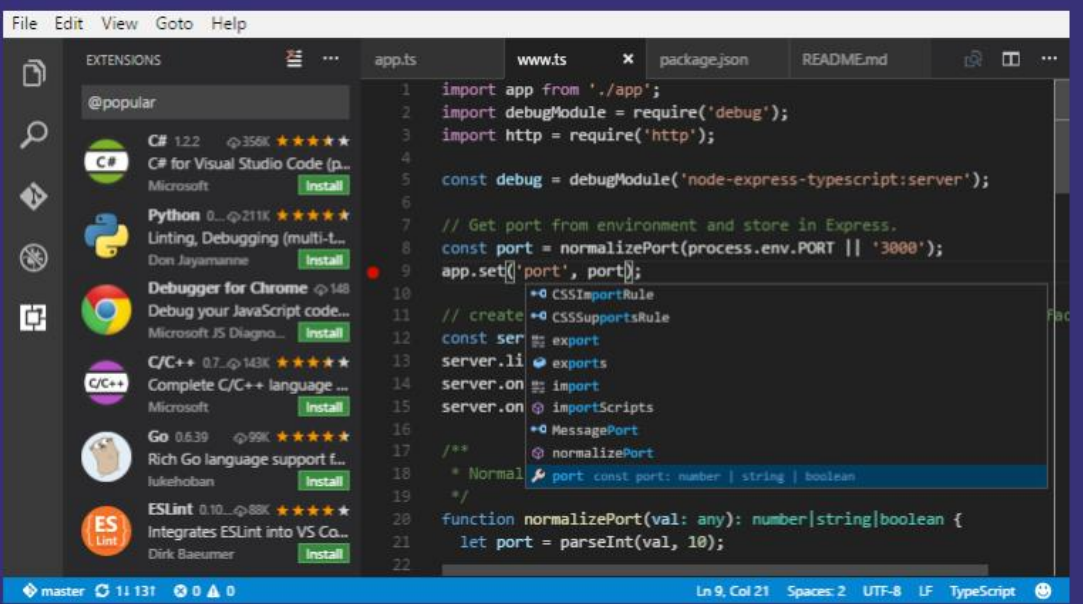
Code editing.
Redefined.

Free. Open source. Runs everywhere.

Download for Windows
Stable Build

Other platforms and Insiders Edition

By using VS Code, you agree to its
license and privacy statement.



IntelliSense



Debugging



Built-in Git

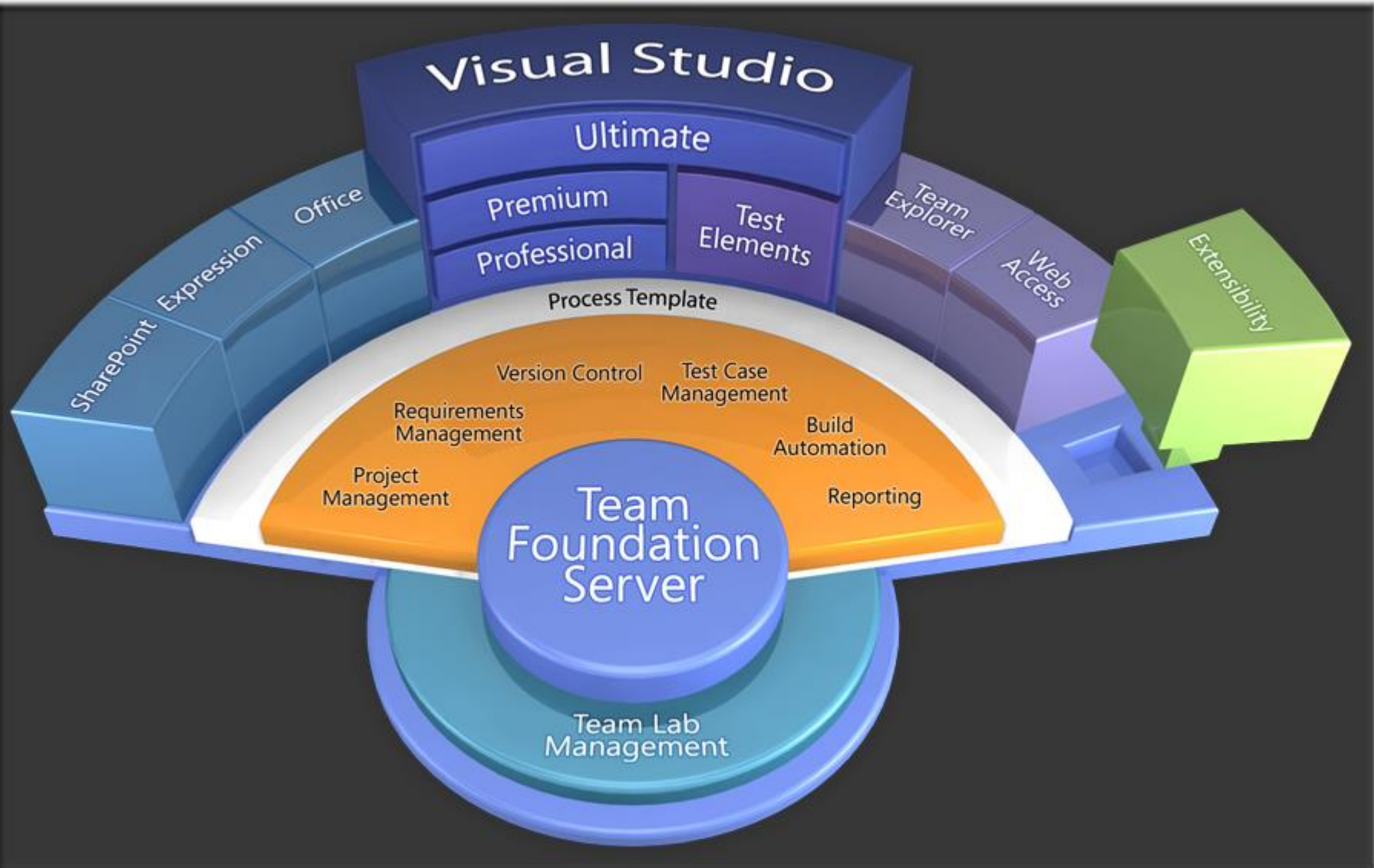


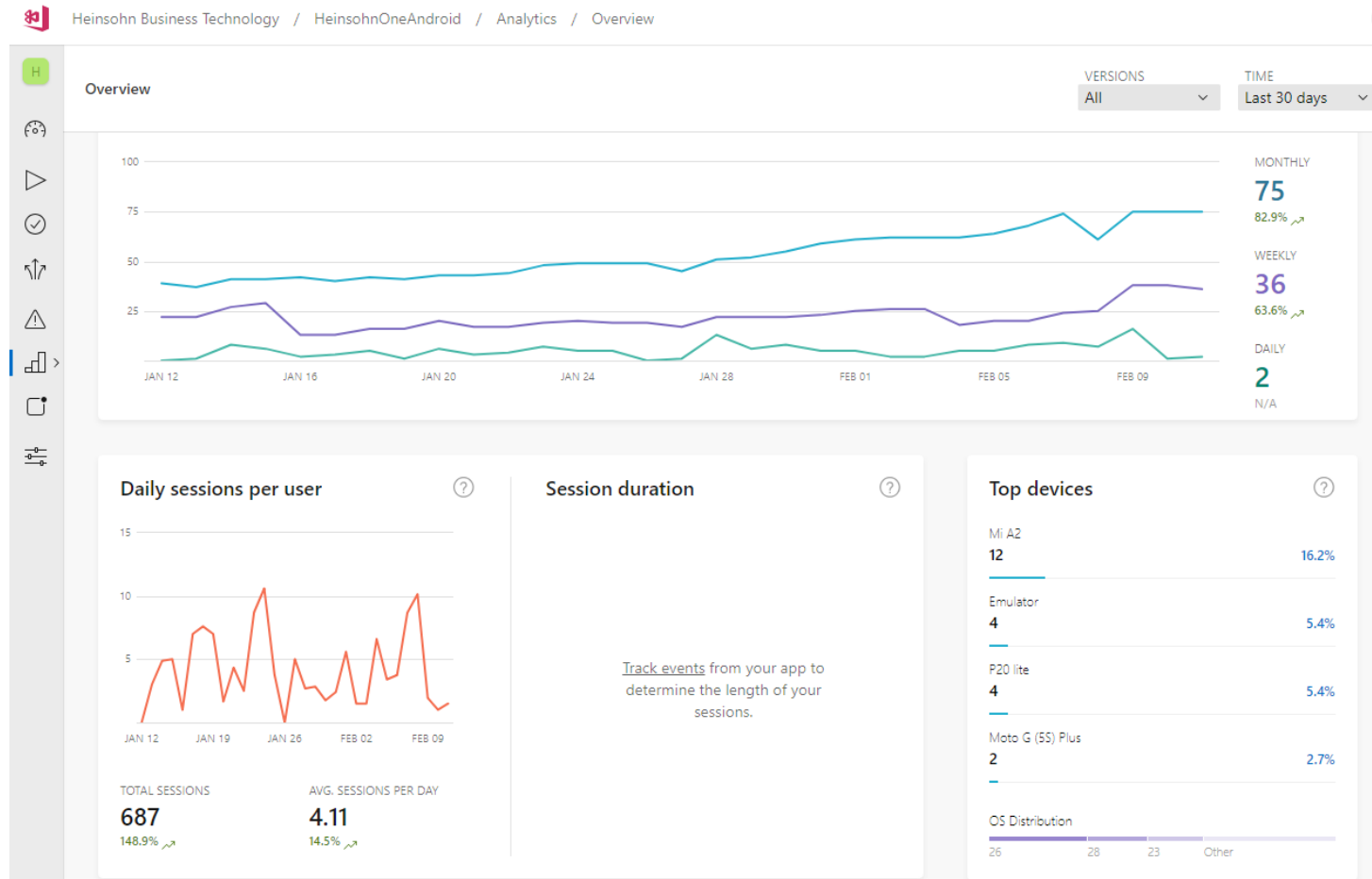
Extensions

<https://code.visualstudio.com>

Team Foundation Services







<https://code.visualstudio.com>



Azure Boards

Ofrezca valor a sus usuarios en menos tiempo usando herramientas Agile probadas para que sus equipos puedan planear el trabajo, debatir sobre él y hacer un seguimiento.

[Más información >](#)



Azure Pipelines

Compile, pruebe e implemente código con CI/CD que funciona con cualquier lenguaje, plataforma y nube. Conéctese a GitHub o a cualquier otro proveedor de GIT y lleve a cabo implementaciones continuas.

[Más información >](#)



Azure Repos

Obtenga un número ilimitado de repositorios GIT hospedados en la nube y colabore para compilar código de más calidad con solicitudes de incorporación de cambios y administración avanzada de archivos.

[Más información >](#)



Azure Test Plans

Pruebe y distribuya soluciones con confianza usando herramientas de pruebas manuales y exploratorias.

[Más información >](#)



Azure Artifacts

Cree, hospede y comparta paquetes con su equipo e incorpore artefactos a sus canalizaciones de CI/CD con un solo clic.

[Más información >](#)

Marketplace de extensiones

Acceda a numerosas extensiones, como Slack, SonarCloud y mil aplicaciones y servicios más creados por la comunidad.

[Más información >](#)

<https://azure.microsoft.com/es-es/services/devops>

.Net Framework Architecture

