

Glosario de términos de Programación para Reclutadores

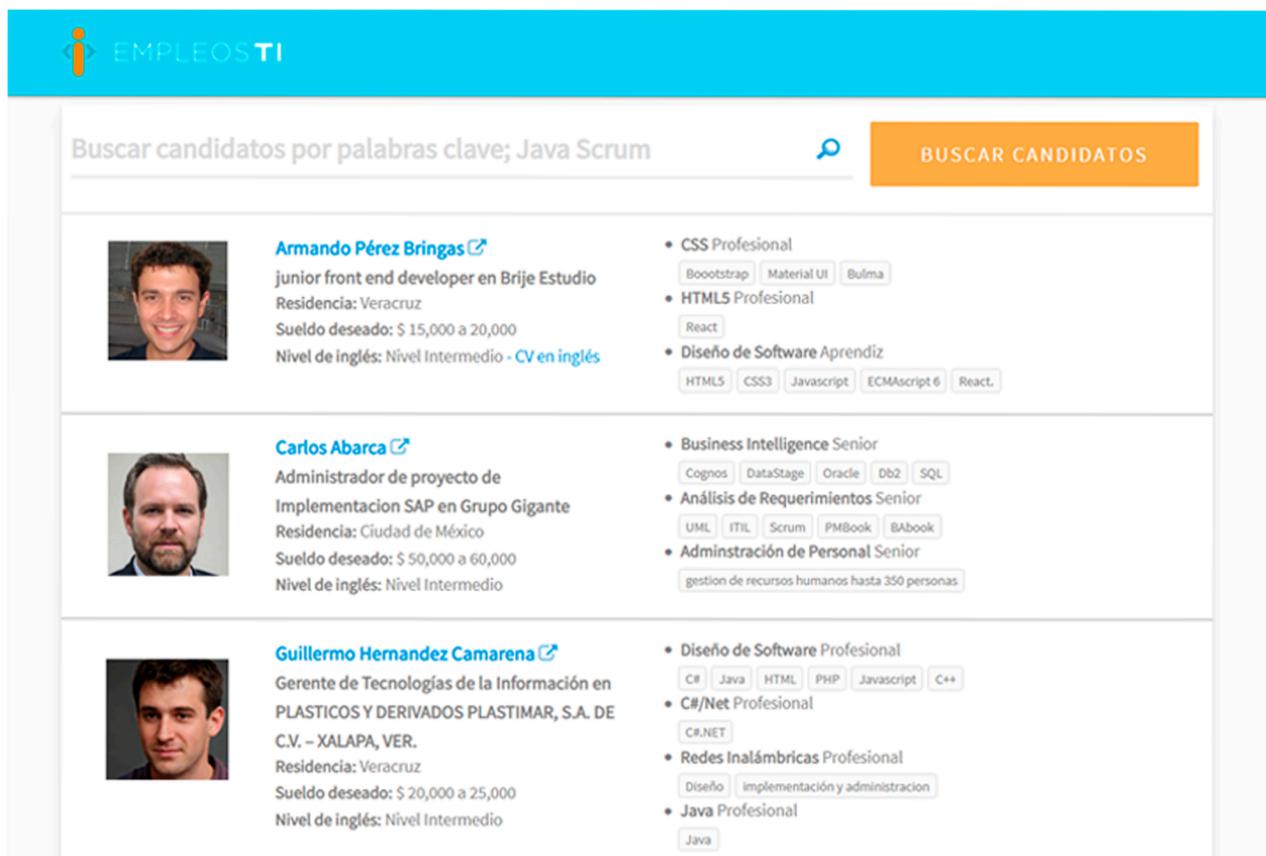


Sobre EmpleosTI

Somos un portal de empleo especializado en perfiles de Tecnologías de la Información. Ayudamos a las empresas a reclutar talento TI en menos tiempo. Dentro de nuestra base de candidatos contamos con más de 200,000 curriculums de tecnología.

Usamos Inteligencia Artificial para mostrarte a los mejores candidatos para tu vacante en cuestión de segundos. Nuestro módulo de IA lee tu vacante y detecta en automático las habilidades requeridas para el puesto. Nuestro sistema te mostrará a los candidatos dentro de nuestra base de talento que mejor cubran las habilidades que buscas en tu vacante.

Conoce nuestros planes de vacantes y membresías en: empleosti.com.mx/reclutador



The screenshot displays the EmpleosTI search interface. At the top, there is a search bar with the text "Buscar candidatos por palabras clave; Java Scrum" and a magnifying glass icon. To the right of the search bar is an orange button labeled "BUSCAR CANDIDATOS". Below the search bar, three candidate profiles are listed:

- Armando Pérez Bringas**: junior front end developer en Brije Estudio. Residencia: Veracruz. Sueldo deseado: \$ 15,000 a 20,000. Nivel de inglés: Nivel Intermedio - CV en inglés. Skills: CSS Profesional, Bootstrap, Material UI, Bulma, HTML5 Profesional, React, Diseño de Software Aprendiz (HTML5, CSS3, Javascript, ECMAscript 6, React).
- Carlos Abarca**: Administrador de proyecto de Implementacion SAP en Grupo Gigante. Residencia: Ciudad de México. Sueldo deseado: \$ 50,000 a 60,000. Nivel de inglés: Nivel Intermedio. Skills: Business Intelligence Senior (Cognos, DataStage, Oracle, Db2, SQL), Análisis de Requerimientos Senior (UML, ITIL, Scrum, PMBook, BAbook), Administración de Personal Senior (gestion de recursos humanos hasta 350 personas).
- Guillermo Hernandez Camarena**: Gerente de Tecnologías de la Información en PLASTICOS Y DERIVADOS PLASTIMAR, S.A. DE C.V. - XALAPA, VER. Residencia: Veracruz. Sueldo deseado: \$ 20,000 a 25,000. Nivel de inglés: Nivel Intermedio. Skills: Diseño de Software Profesional (C#, Java, HTML, PHP, Javascript, C++), C#/Net Profesional (C#.NET), Redes Inalámbricas Profesional (Diseño, implementación y administracion), Java Profesional (Java).

¿Quieres ver cómo funciona EmpleosTI?

[Solicitar demo sin costo](#)

Índice

Términos básicos de programación	4
Lenguaje de programación	4
Código	4
Framework	5
Librería	5
Bug	5
Dominio	6
Hosting	6
Software	6
Cultura	6
Desarrollo Ágil de Software	6
DevOps	7
Scrum	7
Roles	7
Desarrollador Front-End	7
Desarrollador Back-End	8
Desarrollador Full-Stack	8
Desarrollador Móvil	8
Diseñador UI/UX	9
Tester	11
SysAdmin	11
Ingeniero DevOps	11
Administrador de base de datos o DBA (Data Base Administrator)	12
Ingeniero en Ciberseguridad	12
Analista de Sistemas o Business Analyst	12
Project Manager	13
Lenguajes de programación y tecnologías	13
Java	13
C	13
C++	14
C#	14
Python	14
Visual Basic .NET	14
PHP	15
JavaScript	15
Perl	15
Ruby	15
R	15
Go	16
Objective-C	16
SQL	16
Scala	16

Android	16
Bases de Datos	17
Base de Datos	17
Base de Datos Relacional	17
Base de Datos NoSQL	18
Motor de Base de Datos	18
Interfaz de Usuario	19
User Interface	19
Responsive design	19
Usabilidad	19
HTML	19
CSS	19
Material Design	20
Bootstrap	20
Cookies	20
Testing	20
Pruebas de software	20
Prueba automatizada	20
Pruebas unitarias	21
Pruebas de componentes	21
Prueba de humo	21
Prueba de integración	21
Pruebas de regresión	21
Pruebas de aceptación	21
Pruebas de compatibilidad	22
Pruebas de carga	22
Prueba de rendimiento	22
Pruebas de estrés	22
Herramientas de desarrollo y procesos	22
Sistema de control de versiones	22
Git	23
Subversion (SVN)	23
Commit	23
GitHub	23
GitLab	23
Bitbucket	24
IDE	24
Herramientas de compilación	24
Servidor de automatización	24
Integración continua	24
Entrega continua	25
Producción	25
Canalización de Implementación	25
Solicitud / Problema / Incidente	25

Arquitectura	26
Arquitectura	26
API	26
REST	26
SOAP	26
Monolito	26
Sistema distribuido	27
Arquitectura Orientada a Servicios	27
Microservicios	27
Infraestructura	28
Infraestructura	28
Servidor	28
Servicio en la nube	28
SaaS	28
PaaS	29
IaaS	29
Escalabilidad	29
Infraestructura como Código	30
Contenedores	30
Sistema operativo	30
Shell	30
Log	31
Open Source	31
Otros recursos de Reclutamiento TI	32
Curso de Reclutamiento TI	32
Reporte del Mercado Laboral TI 2020	32
Guía de Reclutamiento TI	32

Términos básicos de programación

Lenguaje de programación

Para que una computadora ejecute una instrucción necesitas decirle dicha instrucción en un lenguaje que entienda. A estos lenguajes que la computadora es capaz de entender, se le llaman lenguajes de programación.

Es la herramienta principal que usa un programador para construir programas de computadora como: aplicaciones de celular/tablet, páginas web o programas de escritorio.

Código

Son las instrucciones que escribe un programador en un lenguaje de programación. El código es la etapa donde un programador puede leer y modificar un programa de computadora.

En la siguiente imagen podrás ver un código escrito en el lenguaje de programación Java que le solicita a la computadora mostrar “Hola mundo” en la pantalla.

Código en Java

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

Framework

Un framework proporciona una manera estándar de crear aplicaciones. Consiste de muchas soluciones predefinidas para funcionalidades comunes que son utilizadas para ayudar a crear aplicaciones de software de manera más rápida y estandarizada.

Utilizando un framework, los desarrolladores no tienen que escribir desde cero todas las funcionalidades del software en el que están trabajando porque ya están implementadas en el framework y listas para ser utilizadas con un solo comando.

Un ejemplo muy sencillo de las herramientas que puede ofrecer un framework, son los formularios de inicio de sesión en una página web.

Desarrollar un código desde cero para validar que el correo y contraseña que puso un usuario son correctos, toma tiempo. Un framework puede ayudar a los desarrolladores a no escribir este código y seguir avanzando en otras partes del proyecto de software.

Imagínate los frameworks como una plantilla de excel que ya tiene fórmulas predeterminadas y solo tienes que ingresar algunos datos para obtener el resultado deseado.

Librería

Una librería es una colección de funciones prediseñadas para realizar tareas en específico. Por ejemplo, escribir un código que exporte una base de datos a un excel, le tomaría bastante tiempo a un programador. Por lo que se puede apoyar de una librería que ya tenga desarrollada esta función.

Bug

Un bug es un error en el código de un programa.

Alguna vez haz hecho clic en el botón de una página y no pasa nada, o haz entrado a un sitio web y empieza a parpadear, o el programa simplemente se cierra sin causa aparente.

Estos problemas son causados por errores en el código de un programa, o como le llaman usualmente los desarrolladores: “bugs”.

Dominio

Básicamente es el texto que ingresa una persona en un navegador para ingresar a un sitio web en específico. Por ejemplo el dominio de EmpleosTI es “empleosti.com.mx”.



Hosting

Es el servicio que provee almacenamiento de información, imágenes, vídeo, o cualquier contenido accesible vía Web.

Las empresas usan un servicio de Hosting para tener su sitio web en línea y que los usuarios puedan acceder a él.

Software

En el mundo de la programación se le conoce como software a cualquier programa de computadora. Una aplicación móvil, un sistema operativo, una página web, un programa de escritorio, todo eso se le conoce como software.

Cultura

Desarrollo Ágil de Software

Hace referencia a las metodologías que promueven procesos disciplinados en el desarrollo de software. Estos procesos fomentan el trabajo en equipo, la organización, responsabilidad y mejores prácticas de desarrollo de software. Esto con la finalidad de desarrollar aplicaciones de mayor calidad en el menor tiempo posible.

Algunas de las metodologías de desarrollo ágil de software son: DevOps, Scrum y Kanban.

DevOps

DevOps promueve una colaboración fluida entre el equipo de desarrollo (programadores) y el equipo de operaciones (administradores de servidores, base de datos, testers, etc) para producir productos mejores y más confiables.

Al adoptar una cultura de DevOps junto con prácticas y herramientas de DevOps, los equipos adquieren la capacidad de responder mejor a las necesidades de los clientes, aumentar la confianza en las aplicaciones que crean y alcanzar los objetivos empresariales en menos tiempo.

Scrum

Scrum es una metodología de desarrollo ágil. Durante un sprint (o iteración), los miembros del equipo se auto-organizan para realizar las tareas que les corresponde. Al finalizar el sprint, los miembros del equipo revisan las tareas finalizadas y definen las tareas que se deben de hacer en la siguiente iteración.

A cada tarea de un sprint se le asigna un puntaje para después poder evaluar el rendimiento de los miembros del equipo. De esta manera los managers pueden detectar un bajo rendimiento o predecir un contratiempo antes de que sea demasiado tarde.

Roles

Desarrollador Front-End

El desarrollador Front-End hace todo lo que el usuario ve en la página.

Le da el diseño a las páginas web a través de herramientas como HTML y CSS. Con estas herramientas ellos pueden modificar el color de fondo, los tipos de fuente, el color de las letras o la estructura de texto en una página.

También hacen que las páginas sean dinámicas, o sea que el usuario pueda interactuar con el sitio web. Por ejemplo, cuando estás comprando en línea y agregas a tu carrito un producto, a lado del carrito vas a salir un "1" mostrando que tienes un producto en tu carrito para después comprarlo.



Para lograr que la página sea dinámica los desarrolladores Front-End usan lenguajes de programación como JavaScript. Suelen apoyarse en frameworks como AngularJS, Express, React o Vue.js.

Desarrollador Back-End

A diferencia del Front-End, el desarrollador Back-End crea todos los procesos de código de una página que el usuario no puede ver.

Los desarrolladores Back-End manejan la base de datos de las páginas y la lógica de una página web.

Sigamos con nuestro ejemplo de la tienda en línea. Para que una tienda en línea funcione, el desarrollador Back-End tiene que pensar en toda la lógica del negocio. Tiene que pensar paso por paso cuál va a ser el proceso del usuario. Un usuario necesita crear una cuenta, poder agregar productos en su carrito para después visualizarlos, y debe de ser capaz de realizar un pago con tarjeta de crédito.

Cuando un usuario ingresa los datos de su tarjeta para pagar los productos que seleccionó, la página está corriendo un código que le pide al banco realizar un cobro en la tarjeta. Proceso que el usuario no es capaz de ver y se está ejecutando por medio de un lenguaje de programación, eso es el back-end de una página.

Algunas de los lenguajes más usados por los desarrolladores Back-End son Java, PHP, Python y C#.

Desarrollador Full-Stack

Los desarrolladores Full-Stack son perfiles híbridos que conocen las herramientas que usa el programador Front-End y Back-End.

Pueden hacer desde el diseño de una página web hasta manejar la base de datos y procesos de la página.

Desarrollador Móvil

Los desarrolladores móviles crean apps para celulares o tablets para sistemas operativos como Android o iOS.

Los desarrolladores de aplicaciones para iOS trabajan con el lenguaje de programación Swift. Mientras que los desarrolladores móviles para Android trabajan con Kotlin o Java.

Actualmente se están desarrollando nuevos frameworks para crear aplicaciones que funcionen en ambos sistemas operativos (Android y iOS) como ReactNative. De esta manera los desarrolladores móviles sólo tienen que construir un sólo código.

Diseñador UI/UX

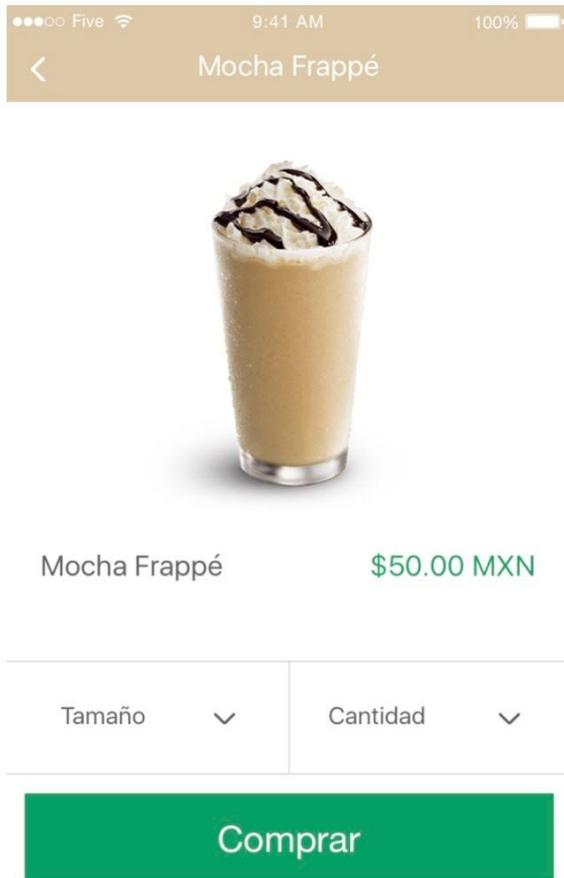
Para entender este perfil, primero debemos de entender qué significan sus acrónimos.

UI significa User Interface (o Interfaz de usuario), es lo que el usuario puede ver en una aplicación. Mientras que UX es User Experience (o Experiencia de Usuario) y es lo que el usuario siente cuando usa una aplicación.

Un UI Designer diseña cómo se va a ver la página web o app en todos los dispositivos como computadora, celular y tablet. Son perfiles con bases fuertes en diseño, hacen que la aplicación se vea bien. Ellos deciden dónde va a ir la barra de búsqueda, donde van a estar los botones, el color de fondo de una sección, los iconos que se van a usar en la página, el color de los enlaces, entre otros elementos de diseño.

Por otro lado, el UX Designer se encarga de crear una buena experiencia para el usuario a través del diseño de la aplicación.

Para explicar mejor cuál es el trabajo de un UX, pondremos de ejemplo la siguiente imagen con dos versiones de una app.



En la primera versión tenemos el diseño de una app que se ve atractiva y cumple con su propósito. El usuario puede seleccionar el tamaño de su café y decidir cuántos quiere comprar. Este sería el trabajo de un UI.

En la segunda versión podemos ver la misma app pero con un diseño que aporta más valor al usuario y mejora su experiencia usando la plataforma. En esta versión, el usuario puede leer la descripción del producto y agregarlo a favoritos para encontrarlo más rápido en el futuro. Este es el trabajo del UX.

Aunque el UI y UX Design son diferentes, la mayoría de las empresas lo tratan como un mismo rol. Basta con meterte a un portal de empleo para encontrar decenas de vacantes con el título UI/UX.

El UI/UX le pasa al Front-End un mockup o diseño de cómo se tiene que ver la aplicación. El Front-End, se encarga de poner el diseño del UI/UX en código. Sin embargo, algunas empresas pueden buscar un perfil híbrido que complemente las dos posiciones, alguien que

cree el diseño visual de la página y que también pueda pasarlo a código a través de HTML, CSS y JavaScript.

Los UI/UX usan herramientas de prototipado para hacer mockups del diseño de una app o página web, como: Invision, Adobe Experience Design, Origami Studio o Sketch.

Tester

El Tester o QA (Quality Assurance) planifica y lleva a cabo pruebas de software para comprobar si un programa de computadora funciona correctamente. Buscan riesgos de sufrir errores de un software, detectan incidencias y los comunican al equipo de programadores. Evalúan el funcionamiento general del software y sugieren formas de mejorarlo.

Todo esto con el objetivo de reducir las posibilidades de errores en el funcionamiento de un programa cuando el usuario final lo utilice.

SysAdmin

Son responsables del mantenimiento, la configuración y el funcionamiento confiable de los sistemas y servidores informáticos. Instalan hardware y software y participan en la investigación y el desarrollo para mejorar continuamente y mantenerse al día con las necesidades comerciales de TI de su organización. L

Los administradores de sistemas también resuelven activamente problemas y problemas con los sistemas informáticos y de servidores para limitar las interrupciones del trabajo dentro de la empresa.

Ingeniero DevOps

Debido a la necesidad de las empresas de mejorar sus procesos de desarrollo de software y operaciones de TI, se creó una nueva posición que su labor sea asegurar la implementación de la metodología DevOps dentro de las empresas.

A estos profesionales se les conoce normalmente como “Ingeniero DevOps” o a veces simplemente “DevOps”.

El ingeniero de DevOps trabaja en conjunto a los desarrolladores de software, y los miembros de operaciones para facilitar desde el desarrollo de software hasta su implementación dentro del sistema.

Por lo cual su principal trabajo será supervisar la integración continua y distribución continua del software.

Administrador de base de datos o DBA (Data Base Administrator)

Hoy en día las empresas grandes generan millones de datos. Una empresa puede generar información de sus productos vendidos, productos comprados, proveedores y clientes. Otro ejemplo serían las páginas web que pueden guardar información de sus usuarios como nombre, apellido, género, edad, correo, etc.

El Administrador de Base de Datos es el encargado de crear una base de datos en la que las empresas puedan guardar toda la información que generan. Los DBA son perfiles solicitados mayormente por empresas que ya generan millones de datos al mes y requieren más estabilidad e integridad en la información que guardan. En empresas pequeñas o startups que apenas generan miles de datos al mes, la base de datos es generalmente manejada por un desarrollador de software.

Para manejar una base de datos efectivamente, los DBA necesitan poder ingresar nueva información, borrar datos ya no necesarios, consultar los datos en cualquier momento, crear respaldos y dar acceso a otros usuarios para manejar la base de datos.

Ingeniero en Ciberseguridad

Un Ingeniero en Ciberseguridad se encarga de la privacidad y protección de datos de las empresas y las organizaciones para hacer frente a los ciberataques. Por lo que su principal labor es detectar fallas de seguridad y habilitar mecanismos para evitar posibles ataques a los puntos vulnerables.

El Especialista en Ciberseguridad es un profesional experto en la seguridad informática, ya que lidian con la protección de la información en redes informáticas, servidores en línea o cloud servers, software de pago, dispositivos móviles, entre otros.

Analista de Sistemas o Business Analyst

El Analista de Sistemas es la conexión entre el cliente que solicita un programa y el equipo de tecnología que lo va a desarrollar.

Analiza el programa que el cliente solicita, determina si el proyecto es viable con las tecnologías disponibles y le comunica al equipo de desarrollo los requerimientos con los que debe cumplir el programa. Además se asegura que el programa alcance las expectativas del cliente.

Los Analistas de Sistemas tienen conocimientos fuertes en software para evaluar la viabilidad de un proyecto y comunicar de manera fácil a los desarrolladores lo que debe de hacer el programa. Sin embargo, su rol no involucra escribir código. Debe de conocer las tecnologías, lenguajes de programación y frameworks con los que trabaja su equipo, para comprender el alcance que pueden tener en los proyectos, pero él no los usa.

Básicamente las habilidades de los Analistas de Sistemas radican en sus soft skills para escuchar lo que quiere el cliente y comunicárselo de manera efectiva al equipo de desarrolladores.

Project Manager

Un Project Manager se dedica a una de las funciones más críticas de las organizaciones, ya que es el responsable de asegurar que los proyectos se realicen en tiempo y forma sin salirse del presupuesto asignado. Son aquellos que monitorean las actividades, asignan tareas, recursos y presupuesto a cada uno de los proyectos de la organización. En lo anterior, se denota con facilidad la importancia de tener un Project Manager para que las actividades de la organización sigan los objetivos y visión de la misma.

Lenguajes de programación y tecnologías

Java

Lenguaje de programación compilado orientado a objetos, similar en sintaxis a C++. Está concebido para dejar a los desarrolladores de aplicaciones “Escribir una vez, ejecutar en todas partes” (write once, run anywhere) lo que significa que un código compilado de Java puede ser ejecutado en todas las plataformas que son compatibles Java sin la necesidad de recompilación.

Java es el rey de los lenguajes de programación. Es un lenguaje multiplataforma que se puede encontrar en miles de programas que usamos día a día. Se utiliza para hacer: programas de escritorio, páginas web y aplicaciones móviles.

C

Lenguaje de programación informático imperativo de uso general que soporta programación estructurada, alcance léxico variable y recursión, mientras que un sistema de tipo estático previene muchas operaciones accidentales.

C fue desarrollado originalmente por Dennis Ritchie entre 1969 y 1973 en Bell Labs, y utilizado para reimplementar el sistema operativo Unix. Desde entonces se ha convertido en uno de los lenguajes de programación más ampliamente utilizados de todos los tiempos.

C++

Lenguaje de programación de uso general. Tiene características de programación genéricas, orientadas al objeto e imperativas y proporciona a las instalaciones una manipulación de memoria de bajo nivel.

C#

Lenguaje de programación que está diseñado para crear una variedad de aplicaciones que corren en el Framework .NET. C# es simple, poderoso, seguro de escribir y orientado a objetos.

Python

Lenguaje de programación que está entre los más populares. (top 3 según un estudio de IEEE Spectrum). Fue lanzado inicialmente en 1991 y ganó popularidad por ser fácil de aprender pero poderoso para resolver problemas. La sintaxis es similar a la familia C++ pero está equipado con una variedad de soluciones modernas.

Es un lenguaje especialmente utilizado en el ambiente de ciencia de datos e inteligencia artificial.

Visual Basic .NET

Lenguaje de programación orientado a objetos multi-paradigma implementado sobre el Framework .NET. Microsoft lanzó VB.NET en 2002 como el sucesor de su lenguaje Visual Basic original. A pesar de que la parte “.NET” del nombre fue abandonado en 2005, “Visual Basic [.NET]” es utilizado para referirse a todos los lenguajes Visual Basic lanzados desde 2002 para distinguirlos del Visual Basic clásico. Junto con el Visual C#, es uno de los dos lenguajes principales que se enfocan en el Framework .NET.

PHP

Lenguaje de script ampliamente utilizado de código abierto de uso general que es especialmente adecuado para desarrollo web y puede ser embebido en HTML.

JavaScript

Lenguaje dinámico, sin tipo, basado en un prototipo de alto nivel. Dependiendo del ambiente, JavaScript puede ser interpretado o compilado. Es un lenguaje de programación multi-paradigma, soportando estilos de programación funcionales, imperativos y orientados al objeto. skiller

Perl

Desarrollado originalmente por Larry Wall en 1987 como un lenguaje de script de uso general para Unix para hacer más fácil el procesamiento de reportes. Desde entonces, ha sufrido muchos cambios y revisiones.

Perl 6, que comenzó como un rediseño de Perl 5 en 2000, eventualmente evolucionó a un lenguaje separado. Ambos lenguajes continuaron siendo desarrollados de manera independiente por diferentes equipos de desarrollo y generosamente tomaron prestadas ideas uno del otro.

Ruby

Lenguaje de programación dinámico, reflexivo, orientado a objetos de uso general. Fue diseñado y desarrollado a mediados de los 90's por Yukihiro "Matz" Matsumoto en Japón.

Swift

Es un lenguaje de programación compilado, multi-paradigma y de uso general desarrollado por Apple Inc. para iOS, macOS, watchOS, tvOS, y Linux.

R

Lenguaje de programación de código abierto y un ambiente de software para computación estadística y gráfica. Es ampliamente utilizado entre estadísticos y mineros de datos para desarrollar software estadístico y análisis de datos.

Go

Lenguaje de programación gratuito y de código abierto creado por Google.

Objective-C

Lenguaje de programación orientado a objetos de uso general que añade la mensajería estilo Smalltalk al lenguaje de programación C. Fue el principal lenguaje de programación utilizado por Apple para los sistemas operativos OS X e iOS, y sus respectivas interfaces de programación (APIs) Cocoa y Cocoa Touch antes de la introducción de Swift.

SQL

Lenguaje de programación para manejo base de datos más popular. Históricamente, este paradigma de programación declarativo ha sido la característica principal para preguntas ad-hoc operadas para introspección de datos ejecutado por usuarios humanos directamente con SQL (en lugar de con un UI). Hoy en día, SQL está también embebido en otros lenguajes de programación de uso más general como Java para acceder a los datos desde bases de datos centrales.

Scala

Lenguaje de programación de uso general que proporciona soporte general para programación funcional y un sistema de escritura estática fuerte. Diseñado para ser conciso, muchas de las decisiones de diseño de Scala apuntan a tratar las críticas de Java.

Android

Sistema operativo más popular del mundo dedicado principalmente a dispositivos móviles. El código fuente es desarrollado por Google bajo el Proyecto de Código Abierto Android

(AOSP, Android Open Source Project). Usualmente las versiones superiores son lanzadas anualmente y son anunciadas en las conferencias de Google I/O. skiller

Bases de Datos

Base de Datos

Base de información almacenada y utilizada por un programa. Las bases de datos tradicionales están organizadas por campos, registros y archivos.

Las bases de datos resguardan información clave para el negocio. Por ejemplo, las grandes tiendas de autoservicio necesitan y utilizan una base de datos para saber cuántos productos tienen en almacén, cuáles son los más vendidos, los horarios con más flujo de ventas, etcétera.

Base de Datos Relacional

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas.

Por ejemplo supongamos que en una tabla tenemos la información de nuestros clientes. Cada cliente tiene su propio nombre, teléfono, correo y dirección. Y cada cliente tiene un vendedor. Todos los vendedores tienen un nombre, teléfono, email, edad etc.

En una base de datos relacional, podemos establecer que hay una relación entre los clientes y vendedores.

Abajo podrás ver una representación visual de cómo funcionan las bases de datos relacionales.



Base de Datos NoSQL

Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala.

Motor de Base de Datos

Las bases de datos no son como un excel que un usuario puede manejar de manera sencilla para modificarla.

Para acceder a la información de una base de datos y modificarlo como sea necesario se usan los motores de base de datos.

Con los motores de base de datos se pueden crear, consultar, actualizar y borrar información de una base de datos.

Algunos ejemplos de motores de base de datos son: MySQL, PostgreSQL o SQLite.

Interfaz de Usuario

User Interface

Es la parte visual de un programa. Es lo que ve el usuario, y con lo que interactúa

Responsive design

Es la filosofía de diseño y desarrollo web que indica que una página o aplicación web se debe de adaptar al dispositivo que usa el usuario.

De tal manera que no importa si estás viendo una páginas web en laptop, PC, tablet o smartphone, en todos se ve bien la página web.

Usabilidad

Básicamente es qué tan fácil es usar una aplicación. Por ejemplo, si una aplicación es fácil de usar y no se necesita de un manual para poderla usar, sino que intuitivamente el usuario sabe cómo debería de manejar la aplicación, entonces se diría que esta aplicación tiene una buena usabilidad.

HTML

Es la herramienta que le ayuda a los desarrolladores web a dar estructura al texto de una página. Con esta herramienta los programadores pueden decidir qué texto va en negritas, itálica o subrayado, crear listados o tablas, insertar vínculos en un texto, entre varias cosas más.

No es un lenguaje de programación, porque no sirve para dar una serie de instrucciones a una computadora. Simplemente da formato a los textos y su estructura.

CSS

CSS es una herramienta para darle estilo y diseño a una página web. Con CSS los programadores pueden definir el color o tamaño de letra, el margen de separación de una sección con otra, el tipo de fuente, el color de fondo o crear botones con diseño.

Como es un una herramienta que solo da estilo al texto de una página, tampoco es considerado lenguaje de programación.

Material Design

Es la filosofía de diseño conceptual de Google que describe cómo es que las aplicaciones deben verse y trabajar. Analiza todo desde animación y estilo hasta diseño gráfico y da orientación sobre patrones, componentes y usabilidad.

Bootstrap

Es un framework web front-end de código abierto gratuito para diseñar sitios web y aplicaciones web. Hace que crear sitios web sea muy fácil, empodera a los diseñadores para seleccionar de una gran colección de elementos pre-creados, comportamientos y atajos. El objetivo es unificar el diseño y permitirles a los diseñadores no técnicos y técnicos mejorar la calidad de su diseño.

Cookies

Las cookies permiten a los sitios web trackear las acciones de sus usuarios. De esta manera los desarrolladores web pueden saber si un usuario ya visitó anteriormente un sitio, en que navegador (Chrome o Microsoft Edge) está viendo el sitio, o también desde qué tipo de dispositivo (pc, celular o tablet) el usuario está viendo la página.

Testing

Pruebas de software

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su puesta en producción. Existen varios tipos de prueba que podría implementar un tester para comprobar el correcto funcionamiento de un programa, a continuación vas a encontrar algunas de las pruebas que existen.

Prueba automatizada

Las pruebas automatizadas son pruebas que se han automatizado a través de herramientas de software y que se pueden ejecutar sin intervención humana.

Pruebas unitarias

Una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código.

Pruebas de componentes

Su objetivo es verificar las funcionalidades y/o usabilidades de los componentes.

Prueba de humo

Se realizan para verificar si las funcionalidades más significativas de la aplicación funcionan o no.

Prueba de integración

Es uno de los tipos de prueba funcional más común y se realiza de forma automatizada. Se realizan para probar componentes individuales con el objetivo de verificar cómo los módulos, que trabajan de forma individual, funcionan cuando están integrados.

Pruebas de regresión

El objetivo de estas pruebas es encontrar errores que puedan haber sido introducidos accidentalmente en la compilación existente.

Pruebas de aceptación

Los usuarios reales usan el software para verificar que cumpla con las tareas requeridas en un ambiente 'real'.

Pruebas de compatibilidad

Sirven para verificar el funcionamiento de un sistemas en distintos entornos. Por ejemplo probar una aplicación web en distintos navegadores como Chrome o Microsoft Edge.

Pruebas de carga

Sirven para validar que la respuesta de una aplicación sea buena cuando está siendo usada por varios usuarios de forma simultánea. Por ejemplo, probar la aplicación con 100 usuarios.

Prueba de rendimiento

Calcular la respuesta a la aplicación con diferentes números de usuario. Probar la aplicación con 1, 10 o 100 usuarios.

Pruebas de estrés

Comprobar cuántos usuarios puede soportar una aplicación de manera simultánea sin presentar errores.

Herramientas de desarrollo y procesos

Sistema de control de versiones

Es la gestión de cambios de los documentos, programas informáticos, sitios web grandes y otras recopilaciones de información.

Los cambios son identificados usualmente por un código de número o letra, expresado como “número de revisión”, “nivel de revisión”, o simplemente “revisión”. Por ejemplo, un juego inicial de archivos es “revisión 1”. Cuando se realiza el primer cambio, el juego resultante es “revisión 2”, y así.

Cada revisión está asociada con una marca de fecha y la persona que hizo el cambio. Las revisiones pueden ser comparadas, restauradas, y con algunos tipos de archivos, fusionadas.

Git

Es un sistema de control de versiones distribuidas de código abierto gratuito para rastrear los cambios en archivos de computadora y coordinar el trabajo en esos archivos entre múltiples personas. Es utilizado principalmente para la gestión de código abierto en el desarrollo de software, pero puede ser utilizado para hacer seguimiento de los cambios en cualquier juego de archivos.

Subversion (SVN)

Sistema de control de versiones de código abierto popular. En contraste con Git, todos los desarrolladores utilizan un repositorio centralizado.

Commit

En los sistemas de control de versiones, un commit es un cambio guardado hecho al código fuente. Los commits son usualmente archivos o directorios añadidos/eliminados, contenidos de archivos modificados, etc.

GitHub

Es un servicio de alojamiento popular de Git. Es usado principalmente para programar. Además de Git, proporciona varias características de colaboración tales como seguimiento de errores, solicitudes de funciones, gestión de tareas y wikis.

GitLab

Justo como GitHub es un alojamiento de repositorio de Git que proporciona wikis y la funcionalidad de seguimiento de problemas. Es un proyecto de código abierto desarrollado por GitLab Inc. Sitio web: [GitLab.com](https://gitlab.com).

Bitbucket

Es un servicio de alojamiento de repositorios Git y Mercurial de propiedad de Atlassian. Como sus contrapartes, proporciona varias características de colaboración tales como seguimiento de problemas y wikis. Sitio web: [Bitbucket.org](https://bitbucket.org)

IDE

“Entorno de Desarrollo Integrado”, es una aplicación de editor de código diseñada para ayudar a los programadores a escribir, ejecutar y depurar códigos. Ejemplos: Eclipse, IntelliJ IDEA, Visual Studio, PHP Storm, etc.

Herramientas de compilación

Son utilizadas para convertir código de programación escrito por el desarrollador a código binario que es ejecutable por una computadora y correrlo. Usualmente maneja las dependencias externas (bibliotecas o frameworks) las cuales son utilizadas en el proyecto. Ejemplos: Gradle, Maven, Rake, MSBuild, Phing.

Servidor de automatización

Ayuda a automatizar la parte no humana del proceso de desarrollo de software con integración continua y facilitando aspectos técnicos de entrega continua. Ejemplos: Jenkins, Bamboo o TeamCity.

Integración continua

La integración continua es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas.

Con la integración continua, los desarrolladores envían los cambios de forma periódica a un repositorio compartido con un sistema de control de versiones como Git. Antes de cada envío, los desarrolladores pueden elegir ejecutar pruebas de unidad local en el código como medida de verificación adicional antes de la integración. Un servicio de integración continua crea y ejecuta automáticamente pruebas de unidad en los nuevos cambios realizados en el código para identificar inmediatamente cualquier error.

Entrega continua

La entrega continua es una práctica de desarrollo de software mediante la cual se preparan automáticamente los cambios en el código y se entregan a la fase de producción. Fundamental para el desarrollo de aplicaciones modernas, la entrega continua amplía la integración continua al implementar todos los cambios en el código en un entorno de pruebas o de producción después de la fase de compilación. Cuando la entrega continua se implementa de manera adecuada, los desarrolladores dispondrán siempre de un artefacto listo para su implementación que se ha sometido a un proceso de pruebas estandarizado.

Producción

Eta en la que un código ya se ve reflejado para los usuarios finales.

Canalización de Implementación

Define la secuencia de etapas a pasar hasta que la aplicación es introducida a producción. Separando el ciclo de vida de la implementación en etapas. Recolecta un incremento de confianza, usualmente al costo de tiempo extra. Las etapas iniciales pueden encontrar a la mayoría de problemas dando paso a una retroalimentación más rápida, mientras que las etapas posteriores proporcionan una exploración más lenta y más rigurosa. Las etapas pueden incluir creación, despliegue, prueba, archivamiento, etc.

Solicitud / Problema / Incidente

Es un informe sobre un problema en particular, su estado y otros datos relevantes dentro de un sistema de seguimiento de problema, Son creados normalmente en un ambiente de servicio de asistencia o servicio de atención al cliente y casi siempre tienen un número de referencia único, también conocido como caso, problema o número de registro de llamada el cual es utilizado para permitirle al usuario o personal de ayuda a ubicar, añadir o comunicar rápidamente el estado del problema o solicitud del usuario.

Arquitectura

Arquitectura

En tecnología de la información (especialmente computadoras y redes más recientes) la arquitectura es un término aplicado al proceso y al resultado de planear y especificar una estructura general, componentes lógicos e interrelaciones lógicas de una computadora, su sistema operativo y su red.

API

Las API 's permiten la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

Por ejemplo, Paypal pone a disposición de los desarrolladores una API para acceder a sus funcionalidades de pago en línea. De tal manera que un desarrollador puede conectar un e-commerce con Paypal, para que sus usuarios puedan realizar pagos dentro de su sitio. Así el programador usa la aplicación de otra empresa para que su sitio acepte pagos y no se tiene que preocupar por desarrollar el mismo.

REST

Es una técnica de arquitectura de software usada para construir APIs que permitan comunicar a nuestro servidor con sus clientes usando el protocolo HTTP mediante URIs lo suficientemente inteligentes para poder satisfacer la necesidad del cliente.

SOAP

Simple Object Access Protocol (Protocolo de Acceso de Objeto Simple) es una manera de intercambiar información estructurada entre computadoras. Comparada con REST, es más formal, menos flexible y requiere una definición de esquema (WSDL).

Monolito

Es un estilo de arquitectura basado en funcionamiento sobre una sola capa de aplicación que tiende a atar juntas a todas las funcionalidades necesarias para la arquitectura. Para

ponerlo simple, significa estar compuesto todo en una sola pieza. El software monolítico está diseñado para ser auto-contenido; los componentes del programa están interconectados e interdependientes en lugar de estar acoplados ligeramente. Además, si algún componente del programa debe ser actualizado, toda la aplicación debe ser reescrita, mientras que en aplicaciones modulares, cualquier módulo separado (tal como un micro-servicio) puede ser cambiado sin afectar otras partes del programa.

Sistema distribuido

Es un modelo en el cual componentes ubicados en computadores en red se comunican y coordinan sus acciones pasándose mensajes, pareciéndole a sus usuarios un sistema coherente y único. Los componentes interactúan entre ellos para lograr una meta común. Las tres características de los sistemas distribuidos son: concurrencia de componentes, ausencia de un reloj mundial y falla independiente de sus componentes.

Arquitectura Orientada a Servicios

Es un estilo de diseño basado en componentes de software discretos (servicios) que colectivamente proporcionan funcionalidades de arquitecturas de software más grandes. Un servicio es una unidad discreta de funcionalidad que puede ser accedida remotamente y puesta en práctica y actualizada de manera independiente, tal como obtener en línea un estado de cuenta de tarjeta de crédito. En este enfoque, los servicios son proporcionados a otros componentes por componentes de aplicación, a través de un protocolo de comunicación sobre una red. El principio básico de la arquitectura orientada a servicios es la independencia de proveedores, productos y tecnologías. La arquitectura orientada a servicios ha sido utilizada y enfocada principalmente en empresas de gran escala.

Microservicios

Una variante del estilo arquitectónico de la arquitectura orientada a servicios (SOA). La idea detrás de los microservicios es que algunos de los tipos de aplicaciones se vuelven más fáciles de crear y mantener cuando están desglosados en piezas componibles más pequeñas que trabajan juntas.

Cada componente es autónomo, desarrollado separadamente y la aplicación es entonces simplemente la suma de sus componentes constituyentes. Esto es en contraste con una aplicación tradicional “monolítica” desarrollada en una sola pieza. El estilo de los microservicios es utilizado por muchas organizaciones (como Netflix, Uber, Facebook) hoy como un punto de inflexión para lograr un alto grado de agilidad, velocidad de entrega y escala.

Infraestructura

Infraestructura

Se refiere al compuesto de hardware, software, recursos de red y servicios requeridos para la existencia, operación y gestión del ambiente de TI de una empresa. Le permite a una organización entregar soluciones de TI y servicios a sus empleados, socios y/o clientes y es usualmente interna a una organización y desplegar dentro de sus propias instalaciones.

Servidor

Un servidor es un equipo que tiene instalado un software que sirve recursos. Esos recursos pueden ser almacenamiento de archivos, almacenamiento web o de email.

De una manera muy simplista, un servidor es una computadora que almacena información para que otras computadoras puedan solicitar acceso a dicha información. Por ejemplo cada vez que entras a una página web, tu computadora solicita acceso a la información de esta página a un servidor.

Servicio en la nube

Los servicios en la nube son servicios que se utilizan a través de Internet. Es decir, no están físicamente instalados en tu computadora.

Antes de que apareciera la nube, todos los programas se tenían que instalar en tu propia computadora. Los servicios en la nube son programas que se alojan en un servidor accesibles desde cualquier dispositivo conectado a Internet.

Los servicios en la nube se pueden catalogar en Software as a Service (SaaS), Platform as a Service (Paas) y Infrastructure as a Service (IaaS).

SaaS

Software como Servicio (SaaS, Software as a Service), es un modelo de entrega y licencias de software sobre una base de suscripción y es alojado centralmente. SaaS es la forma

más familiar de servicio en la nube para los consumidores. Los proveedores de SaaS hacen que la aplicación esté disponible a todos los usuarios a través del internet. Los clientes de SaaS pueden disfrutar del software sin tener que preocuparse del desarrollo, mantenimiento, soporte, actualizaciones o copias de respaldo. La desventaja, sin embargo, es que su experiencia de software depende del proveedor de SaaS, el cual es responsable por la estabilidad, reporte, facturación y seguridad.

Ejemplos de SaaS son: Gmail, Dropbox, Salesforce, o Netflix.

PaaS

Plataforma como Servicio (PaaS, Platform as a Service) es una categoría de servicios de cómputo en la nube. PaaS funciona a un nivel menor que SaaS, proporcionado típicamente una plataforma sobre la cual el software puede ser desarrollado y desplegado. Los proveedores de PaaS abstraen mucho del trabajo de lidiar con los servidores y le dan a los clientes un ambiente en el cual se hicieron cargo del sistema operativo y el software del servidor, así como también el hardware del servidor subyacente y la infraestructura de red, dejando libres a los usuarios para enfocarse en el aspecto empresarial de la escalabilidad y en el desarrollo de la aplicación de su producto o servicio. Les permite a los clientes desarrollar, ejecutar y gestionar las aplicaciones sin la complejidad de crear y mantener la infraestructura típicamente asociada con el desarrollo y lanzamiento de una aplicación.

PaaS hace el desarrollo, prueba y despliegue de las aplicaciones de manera rápida, simple y rentable. Ejemplos de PaaS: Pivotal Cloud Foundry, Heroku.

IaaS

Infraestructura como Servicio (IaaS, Infrastructure as a Service) es el modelo de servicio en la nube más básico que ofrece infraestructura informática – máquinas virtuales y otros recursos – como un servicio para los suscriptores. Es un nivel más bajo comparado con PaaS. Típicamente, IaaS proporciona hardware, almacenamiento, servidores y espacio de centro de datos o componentes de red. Permite ampliar y reducir rápidamente con demanda y pago solamente por lo que utilice. Esto hace de IaaS adecuada para cargas de trabajo temporal, experimental o que cambian inesperadamente. Ejemplos de IaaS: Amazon Web Services y su EC2.

Escalabilidad

Es la capacidad de un sistema de soportar un número de usuarios cada vez mayor, sin perder calidad en los servicios.

De tal manera que si una aplicación es utilizada por 10,000 o 100,000 personas, la aplicación sigue funcionando correctamente para todos sus usuarios.

Infraestructura como Código

IaC, Infrastructure as Code, también referido como infraestructura programable, significa escribir un código (el cual puede ser hecho utilizando un lenguaje de alto nivel o cualquier lenguaje descriptivo) para gestionar las configuraciones y automatizar el aprovisionamiento de infraestructura en adición a los despliegues. IaC no es solo sobre escribir scripts, también involucra utilizar prácticas de desarrollo de software probadas y demostradas que ya están siendo utilizadas en el desarrollo de aplicaciones, por ejemplo control de versiones, prueba, pequeños despliegues, uso de patrones de diseño. En resumen, esto significa que Ud. escribe un código para aprovisionar y gestionar su servidor, además de automatizar los procesos. Es un enfoque para manejar la infraestructura de TI en la era de la nube, microservicios y entrega continua.

Contenedores

Tal y como los grandes barcos de transporte utilizan contenedores para organizar, acceder y entregar sus mercancías de forma más fácil y eficiente, el mundo del software empresarial (en especial desde la aparición de la nube híbrida) recurrió a una tecnología análoga: la virtualización basada en contenedores, también conocida simplemente como contenedores.

Los contenedores de software son ambientes de ejecución livianos que proveen a las aplicaciones con los archivos, variables y librerías que necesitan para operar. Se utilizan para garantizar que una aplicación se ejecute correctamente cuando cambie su entorno, con una reducción al mínimo de las fallas posibles y una maximización de su portabilidad.

Sistema operativo

Es el software del sistema que maneja el hardware informático y los recursos de software. Osea es el sistema que nos permite acceder y utilizar los programas de una computadora.

Los sistemas operativos son lo que normalmente conocemos como Windows, MacOS, Linux, iOS o Android.

Shell

Interfaz que permite al usuario interactuar con una computadora. Proporciona una interfaz de usuario para acceder a los servicios del sistema operativo. En general, los shells de los sistemas operativos utilizan la interfaz de comando de línea (CLI) o la interfaz gráfica de

usuario (GUI), dependiendo del rol de la computadora y de la operación particular. El nombre viene del hecho que los shells (cáscaras) son capas alrededor del núcleo del sistema operativo.

```
chris@ubuntu:~$ bash
chris@ubuntu:~$ zsh
chris@ubuntu ~ % tcsh
ubuntu:~> dash
$
```



Log

Es un registro de lo que ha pasado. Típicamente ayuda a diagnosticar problemas u obtener ciertos entendimientos sobre qué es lo que pasa en el ciclo de vida de una aplicación.

Open Source

Código fuente abierto software libre, se refiere a un programa cuyo código fuente está disponible al público general, gratis, para usar y modificar.

Otros recursos de Reclutamiento TI

Curso de Reclutamiento TI

El Curso de Reclutamiento TI es una iniciativa de Empleos TI para ayudar a los profesionales de Recursos Humanos a aprender de una manera rápida y sencilla reclutar talento de tecnología. En este curso aprenderás desde a manejar un lenguaje técnico básico hasta a generar estrategias efectivas de reclutamiento de tecnología.

[Ver más](#)

Reporte del Mercado Laboral TI 2020

Hemos curado los datos de más de 8,900 vacantes especializadas en Tecnologías de la Información publicadas en 2019 , para conocer los skills técnicos más demandados, los perfiles con mayor crecimiento, las ciudades que más requieren profesionales especializados en el sector y en qué meses hay más vacantes publicadas. Asimismo, este año hemos enriquecido el reporte con un análisis de sueldos de perfiles de TI. (Salarios por rol, estado, tecnología y años de experiencia)

[Ver más](#)

Guía de Reclutamiento TI

Emmanuel Olvera, CEO de EmpleosTI, te comparte sus conocimientos de 10 años de experiencia reclutando cientos de personas para perfiles de tecnología como Reclutador Senior en Sigtao Software (Fábrica de Software), Indra Systems (Consultoría especializada en banca) y Resource IT (Partner de Oracle). Más 5 años desarrollando estrategias para atracción de candidatos para Empleos TI, logrando tener más de 100,000 perfiles de tecnología registrado en el portal

[Ver más](#)