



INDUSTRIA 4.0

IaaS, PaaS, SaaS y FaaS



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ
TECNOLOGÍA INFORMÁTICA
ÁREA DE CIENCIAS DE LA COMPUTACIÓN
6 DE MARZO DE 2019

RAÚL MARVÁN MEDINA

Contenido

Industria 4.02

IaaS.....3

 Ejemplos.....4

PaaS.....4

 Ejemplos.....4

SaaS.....4

 Ejemplos.....4

FaaS.....5

 Ejemplos.....5

Referencias.....5

Industria 4.0

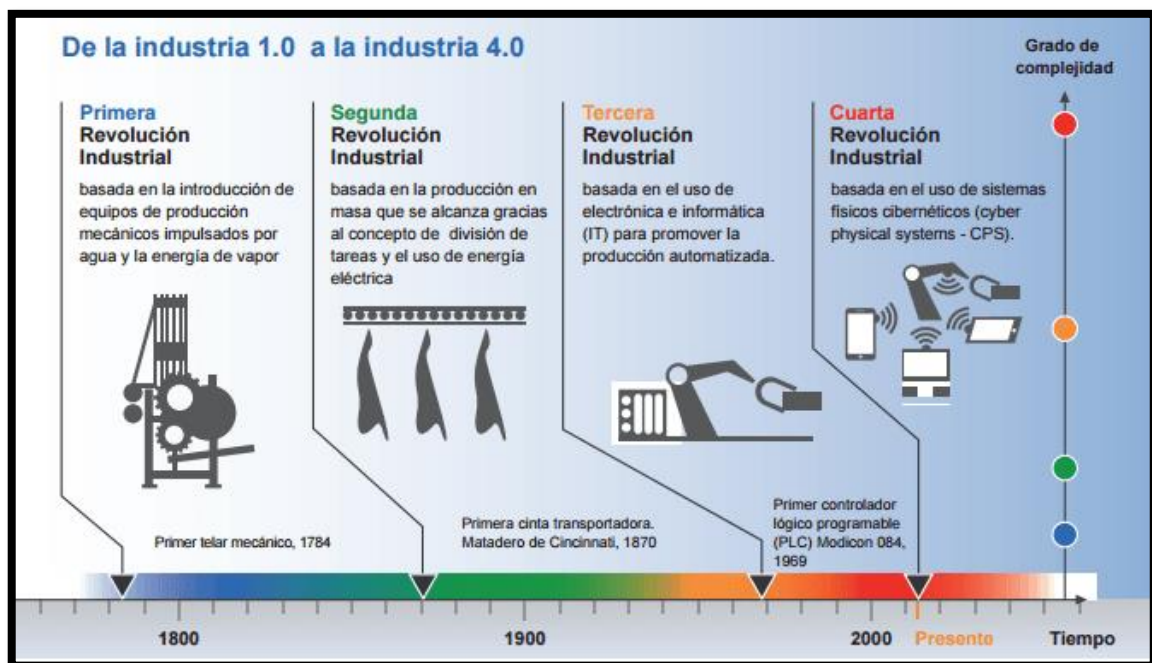
En el año 2013 la consultora Mckinsey &Co adelantaba que la transformación digital de empresas y fabricantes estaba teniendo un impacto mayor en donde el gran público no estaba mirando:

1. En la organización y producción de las fábricas
2. En la gestión de la relación con el cliente

Si durante los pasados años se ha estado centrando la atención en las ventas online, las redes sociales, las APP para móvil y el marketing digital, todo ello desarrollado por gigantes digitales como Amazon, Google, Facebook y Apple.

El concepto de industria 4.0 consiste en la introducción de las tecnologías digitales en las fábricas en general. Es la forma que hay de llamar al fenómeno de transformación digital aplicado a industria de producción.

Si durante años se ha hablado del impacto del Internet de cosas (IoT) en industrias como la energética o de infraestructuras, bajo el concepto de Smart Cities. Ahora toca hablar de “Industria Inteligente” o Industria 4.0.



Lo que ofrece la industria 4.0 a través de la digitalización y el uso de plataformas conectadas es:

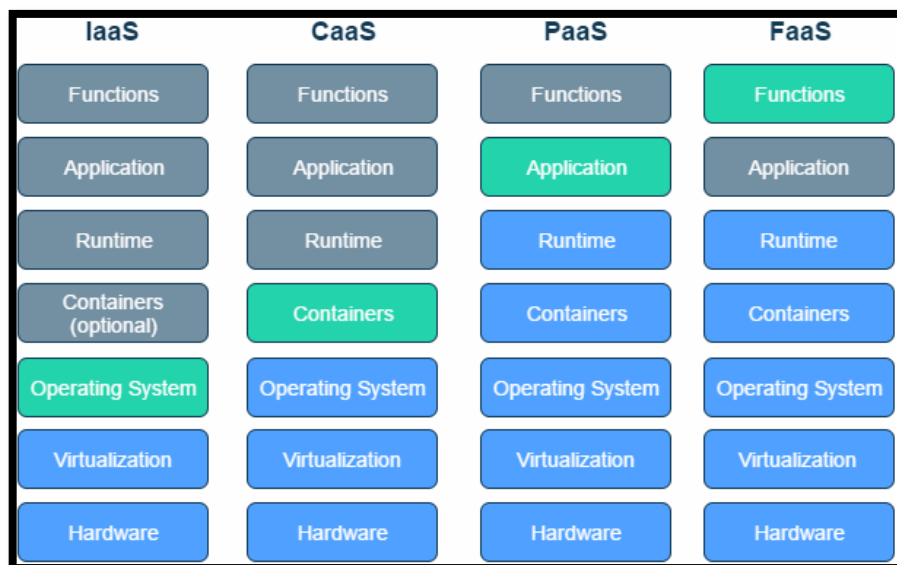
1. Capacidad de adaptación constante a la demanda,
2. Servir al cliente de una forma más personalizada,
3. Aportar un servicio post venta uno a uno con el cliente,
4. Diseñar, producir y vender productos en menos tiempo,
5. Añadir servicios a los productos físicos
6. Crear series de producción más cortas y rentables

7. Aprovechar la información para su análisis desde múltiples canales (Redes sociales, IoT) donde ser capaces de analizarla y explotarla en tiempo real.

La nueva industria 4.0 tiene varios ejes entorno a los que se articula:

1. Big data y análisis de datos
2. **Cloud Computing**
3. Ciberseguridad
4. Robótica
5. Internet de las cosas
6. Simulación y prototipado
7. Realidad aumentada
8. Cultura
9. Integración de procesos

Para esta investigación se abarcarán los temas correspondientes al **Cloud Computing** conocida también como servicios en la nube, informática en la nube, nube de cómputo, nube de conceptos o simplemente "la nube", es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet. Para el resto del documento hablaremos de los servicios que ofrece como **XaaS** (... as a Service).



IaaS

Es el nivel más bajo de todos los XaaS. Nos da una gran potencia pero requiere mucha configuración. IaaS da una máquina virtual que tenemos que mantener. La diferencia entre IaaS y tener una sala de servidores física es que no tenemos que comprar ninguna computadora física y podemos tener servidores en varias partes del mundo. Sin embargo, en comparación con otros XaaS, IaaS es más difícil de mantener y requiere un buen ingeniero que configure las máquinas virtuales para que funcionen de manera eficiente y segura.

Proporciona solo una infraestructura base (Máquina virtual, Software Define Network, Almacenamiento adjunto). El usuario final tiene que configurar y administrar la plataforma y el entorno, implementar aplicaciones en él.

Ejemplos

AWS (EC2), GCP (CE), Microsoft Azure (VM)

PaaS

Es una forma sencilla de implementar una aplicación en una tecnología determinada (por ejemplo, Node.js, Ruby, PHP, Python, Java, .NET). Las plataformas más populares son Heroku y Google App Engine.

El principal inconveniente es que no es demasiado flexible porque es imposible instalar dependencias del sistema personalizadas (por ejemplo, desde apt-get) y puede usar solo una de las tecnologías disponibles. Entonces, si crea su propio lenguaje de programación que no se ejecuta en ninguna plataforma disponible (por ejemplo, en Node.js puede usar TypeScript y CoffeeScript además de JavaScript o en la Máquina Virtual Java, puede usar Scala, JRuby, Jython, Kotlin y Groovy además de Java) no puede usar PaaS.

Otra desventaja es que los datos no son seguros. Si, por ejemplo, utiliza Heroku como PaaS y mLab como DBaaS (Base de Datos como servicio), no solo mLab tiene acceso a sus datos, sino que Heroku también podría hacerlo porque nunca sabe qué código se ejecuta realmente en el servidor. Tal vez no sea su código porque agregan sus propios middlewares para registrar algo.

Proporciona una plataforma que permite al usuario final desarrollar, ejecutar y administrar aplicaciones sin la complejidad de construir y mantener la infraestructura.

Ejemplos

Google App Engine, CloudFoundry, Heroku, AWS (Beanstalk).

SaaS

A veces se lo denomina "software bajo demanda". Por lo general, los usuarios acceden a un cliente ligero a través de un navegador web. En SaaS, todo puede ser administrado por proveedores: aplicaciones, tiempo de ejecución, datos, middleware, sistemas operativos, virtualización, servidores, almacenamiento y redes, los usuarios finales tienen que usarlo.

Proporciona un software listo para usar pero no requiere ninguna implementación / mantenimiento del servidor por nuestra parte.

Ejemplos

Gmail, Google Drive, Dropbox, SparkPost, SendGrid, Slack.

FaaS

Proporciona una plataforma que permite a los clientes desarrollar, ejecutar y administrar las funcionalidades de las aplicaciones sin la complejidad de construir y mantener la infraestructura.

Es incluso más simple que la PaaS. Como su nombre indica, se basa en las funciones que pueden ser activadas por un evento dado, por lo que es una arquitectura basada en eventos. El nivel de simplicidad es tan alto que se denomina arquitectura sin servidor. El desarrollador simplemente escribe una función y no tiene que reflexionar sobre temas como la implementación, los recursos del servidor o la escalabilidad. Es porque FaaS es autoescalable. Por lo tanto, la facturación se basa en el consumo real, no en las necesidades de recursos declaradas.

Uno de los mayores inconvenientes de FaaS son las tecnologías soportadas. Hay incluso menos tecnologías disponibles que con PaaS. Más exactamente (a partir de 2017) AWS Lambda es compatible con .NET, Java, Node.js y Python, Microsoft Azure Functions es compatible con .NET, Java, Node.js y PHP, mientras que Google Functions solo admite Node.js. Como puede observar, ninguno de los FaaS más populares es compatible con Ruby, a diferencia de PaaS (por ejemplo, Heroku).

Ejemplos

AWS (Lambda), Google Cloud Function, Microsoft Azure Functions, Iron.io, Webtask.io.

Referencias

<https://papelesdeinteligencia.com/que-es-industria-4-0/>

<https://medium.com/@nnilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e>

<https://brainhub.eu/blog/cloud-architecture-saas-faas-xaas/>