BBVA
Next Technologies

Radar de Tecnologías 2018

Creando Oportunidades

Introducción

BBVA Next Technologies es una empresa global experta en ingeniería de *software*, formada por más de 1.200 expertos en tecnología, que impulsamos la transformación tecnológica de BBVA. Como compañía especializada en tecnologías emergentes y disruptivas, nuestra vocación es mantenernos siempre un paso por delante, mirando al futuro para anticiparnos a las necesidades y cambios que experimenta la industria. Para ello, contamos en todo momento con los mejores expertos en tecnología y desarrollamos metodologías propias para alcanzar la excelencia e incrementar la capacidad tecnológica del Grupo.

Este es el primer Radar Tecnológico que publicamos como BBVA Next Technologies, compañía creada tras la fusión de las empresas del Grupo BBVA, BEEVA e i4S.

Nuestro propósito es ofrecer una visión panorámica cualitativa y cuantitativa, mediante indicadores definidos a tal efecto, de las herramientas, lenguajes, *frameworks* y metodologías que han sido claves en las áreas en las que trabajamos en BBVA Next Technologies. Este informe nos permite actualizar los principales nodos de conocimiento, apoyando el proceso de investigación y estableciendo el grado de novedad de un desarrollo. De esta forma, orientamos la mejor estrategia de desarrollo tecnológico y respaldamos la toma de decisiones en nuestra estrategia de investigación e innovación. Una tarea apasionante, si tenemos en cuenta el ritmo al que cambia el mundo *tech*.

En los últimos años hemos visto cómo en el mercado se consolidaban algunas tecnologías, como *blockchain*, se estabilizaban otras, como es el caso de *frameworks* como Angular, y algunas alcanzaban la madurez, como la realidad virtual y la realidad aumentada.

Nuestra aproximación a la nube continúa siendo agnóstica, con AWS siendo aún la cloud pública en la que se despliegan la mayor parte de los proyectos big data de la compañía, al mismo tiempo que estamos viendo un gran crecimiento en Google Cloud y Microsoft Azure. Además continuamos desarrollando un amplio expertise en Openstack como tecnología de *cloud* privada.

Este documento está dividido en los diferentes ámbitos en los que trabajamos en la compañía: *backend, blockchain, front*, movilidad,*big data*, IoT, *cloud,* DevOps, *machine learning* o HCI. Este año además hemos incluido el apartado de Ciberseguridad, un área que incorporamos a nuestros ámbitos de expertise tras la reciente fusión con i4S.

Este Radar de Tecnologías es una suerte de brújula para todos los que formamos BBVA Next Technologies, mediante el que nos anticipamos a los cambios que nos trae el futuro para seguir marcando la diferencia en el sector.

BBVA
Next Technologies

índice

Radar de tecnologías 2018

BBVA
Next Technologies

01. Backend

Este año, **BBVA Next Technologies** seguirá manteniendo en su core de lenguajes de programación **Java, Python y Node.js.** Además, este año se añade **Golang como lenguaje troncal**

Estado actual

Este año, BBVA Next Technologies seguirá manteniendo en su *core* de lenguajes de programación **Java**, **Python** y **Node.js**. Además, se añade **Golang** como lenguaje troncal y **Clojure** y **Rust** están siendo explorados a lo largo de 2018.
Aunque se ha avanzado en el uso y adopción de **Scala**, aún no es un lenguaje lo suficientemente adoptado en la compañía como para considerarlo un lenguaje de programación *core*. Al igual que éste, **Groovy** también funciona sobre JVM, y además de compartir similitudes en la sintaxis, ofrece una serie de características como la capacidad de usar DSL, facilidades en su sintaxis similares a Python y hace más sencillo el desarrollo con TDD. Aunque hasta ahora nuestro uso se reduce a facilitar o potenciar nuestros desarrollos con *plugins* para Gradle o Jenkins, confiamos en que esto va ampliarse a otros contextos.
Por otro lado, la simplicidad y claridad que aporta **Kotlin**, junto con la posibilidad de coexistir con JAVA de manera natural, hace que pase de la fase de "Exploración" a la de "Adopción". Como características más destacadas: incluye el control de *null* en su sistema de tipos, permite extender código de terceros (de una manera similar a los prototipos de JavaScript) utilizando "extension functions" y compila a Bytecode y JavaScript.

_Microservicios

En cuanto al campo de los microservicios, BBVA Next Technologies seguirá dedicando esfuerzo al aprendizaje y adopción de tecnologías en un campo que cada vez parece tener más importancia en el sector. La tendencia es la de encontrar una solución madura y eficiente para implementar microservicios en, al menos, los lenguajes de programación considerados como *core*.
El *framework* de **Spring Cloud** se ha consolidado en la compañía como referente en el mundo Java. Este año se ha intentado ahondar un poco más en algunos de los módulos nuevos que han surgido recientemente, como son **Spring Cloud Contract, Spring Cloud Sleuth y Spring Cloud Function**. Igualmente, se seguirán muy de cerca las posibles mejoras que Spring introduzca en este *framework*, como podría ser **Spring Cloud Gateway**, un prometedor sustituto del *router* nativo de la arquitectura de Netflix (Zuul).
En lo que respecta a **Go/Golang, *Go-kit*** parece liderar el segmento de microservicios en este lenguaje, diseñado para trabajar con RPC y con una división clara de responsabilidades que hacen más sencillo el desarrollo.
Respecto a los microservicios políglotas o concernientes a **Node.js** y/o combinaciones con Python, se ha trabajado en proyectos utilizando arquitecturas orientadas a eventos. Para profundizar en este aspecto BBVA Next Technologies asistió a formaciones con RisingStack, empresa referente en el mundo de microservicios y Node.js.
Algunas tecnologías en las que BBVA Next Technologies invertirá esfuerzo para evaluar y explorar en este campo son:

- **Lagom**: *framework* que permite construir microservicios sobre Java/Scala con Akka o Play.
- **Vert.x**: alternativa a Spring Cloud para desarrollar servicios de red (incluyendo HTTP) que además cuenta con un *bus* de eventos y un sistema de actores (Verticles). Está diseñado para ser reactivo y no bloqueante, es una librería más que un *framework* (no instrumenta clases y es más directo que Spring), se puede programar en varios lenguajes (Java, Kotlin, Scala, JavaScript...) y el rendimiento es muy bueno. Es *open source* (fundación Eclipse) y basado en Netty.
- **Nameko**: *framework* para Python que permite la implementación de microservicios que ofrece algunas herramientas como eventos asíncronos o RPC sobre AMQP.

_APIs

En cuanto a APIs, se sigue manteniendo **RAML** como el lenguaje de definición de APIs por excelencia, aunque en paralelo se seguirán de cerca los avances de **Swagger**.
También se seguirá explorando la posibilidad de incorporar librerías **GraphQL** que den más flexibilidad a la implementación de las APIs a través de una sintaxis de *query* propia y estandarizada. También se explorará **RSQL** como mecanismo para hacer más flexible y amigable la manera de incorporar filtros a las URIs.
Por último, los avances en *serverless* por parte de AWS hacen que la implementación de APIs utilizan-

do el conjunto [**Cognito / API Gateway / Lambdas**] sea cada día más plausible y sencillo, a la par que práctico. Por ello, se está intentando explorar fortalezas y debilidades de esta opción.
Por último, parece bastante interesante el uso de **Kong**, un API *manager open source* que ha llamado nuestra atención por su simplicidad y que es lo suficientemente maduro como para que pase a la fase de "Adopción".

_Mensajería

La tecnología considerada como *core* para la empresa es **RabbitMQ**, que es un excelente *broker* de mensajería bastante maduro y que se ha consolidado como indispensable.
Sin embargo, se quieren explorar otras alternativas más recientes, como **Kafka**, que es bastante prometedor, aunque le faltan algunas características que sí que tiene RabbitMQ (como por ejemplo, colas federadas), y que se comporta algo mejor en cuanto al escalado y eficiencia en sistemas relativamente complejos con condiciones de alta carga.
También se explorará **ActiveMQ**, que es otro de los grandes *brokers* de mensajería existentes, con el objetivo de compararlo con **RabbitMQ** e identificar casos de uso en los que se pueda comportar mejor que aquel.

_Frameworks

Spring sigue siendo el *framework* por excelencia dentro de la empresa para JAVA, que en la versión 5.X incorpora en su *core reactive* a Netty. Tampoco se debe perder de vista **Guice**, como alternativa. Otros *frameworks* interesantes que se seguirán de cerca para adoptarlos son **Netty** y **Akka**.
En lo que a Python se refiere, **Flask** se mantiene como el *framework* más utilizado por su potencia y facilidad de uso. Otro interesante que se debería seguir de cerca es **Django**.
Express continúa siendo el *framework* más utilizado en Node.js. Sería interesante explorar otros como puede ser **Loopback**, que parece estar evolucionando.
Por último, en Scala se explorarán el *framework* **Play**, *open source* y reactivo por definición, ya que se construyó sobre Netty; así como **Scalatra**, un *framework* ligero para construcción de aplicaciones web y APIs.

_Servidores de aplicaciones

En cuanto a servidores de aplicaciones, se explorarán alternativas a **Tomcat,** como **Jetty** y **Undertow**. Éste último ya estaba en el horizonte el año pasado, aunque no ha sido lo suficientemente utilizado y por lo tanto se mantiene para explorar.

_Descubrimiento de servicios / Servidores de configuración

Por la parte de *service discovery* adicionalmente a **Consul**, el cual se mantiene como solución robusta, aparece **Linkerd**, no como alternativa, sino como complemento, debido a su gran versatilidad, ya que adicionalmente sus características lo hacen una solución muy completa para tener en cuenta.
Por su parte **conf.d** promete ser una solución totalmente integrable con Consul u otros sistemas, permitiendo realizar cambios en configuración de una forma sencilla y controlada. De momento se están realizando pruebas para ver su aplicabilidad en soluciones más complejas.

_Gestores de dependencias

Aunque **Maven** sigue siendo la herramienta más utilizada con diferencia para la gestión de dependencias en desarrollos Java, ganan fuerza otras alternativas como **Gradle**, algo más flexible y, desde luego, con mejor rendimiento que el primero, aunque aún carece de la gran comunidad de *plugins* que posee Maven. En este punto continuamos ampliando su uso y aprovechando sus características y potencial. Por otro lado, continuamos utilizando **SBT** para proyectos **Scala**, **NPM** para proyectos en **Node.js** y **Pip** con ***setuptools*** para **Python**.
Se utiliza el gestor de paquetes **Yarn** sobre **NPM** porque bloquea las versiones de los paquetes y sus dependencias y realiza un cacheo de estos.
Para el desarrollo con **Go**, creemos que **Glide** va a coger más fuerza frente a otros gestores de dependencias, aunque **Godep** sigue aún muy en uso por proyectos de relevancia como **Kubernetes**.
A su vez, se comenzarán a explorar otros como **Bazel** para varios lenguajes.

_Tests

Para JAVA, las herramientas *core* para *testing* unitarios y de integración siguen siendo **JUnit** y **Mockito.** Otras herramientas recientes para Mocking como **Wiremock** pasan a la fase de adopción este año.
Incluiremos en el radar **Spock**, que corre sobre JUnit, y nos facilita realizar tests más expresivos y cortos para Java y Groovy.
Adicionalmente, en el caso de Golang seguimos utilizando **GinkGo/Gomega** y **Testify** para tests de aceptación y unitarios respectivamente, entre el gran universo de herramientas de *testing* existentes para este lenguaje.
Respecto a Python, se sigue apostando por **Pytest** para los tests unitarios, complementándolo con la librería **Coverage** para verificar la cobertura de código alcanzada y cumpliendo con el estándar de calidad propuesto en el PEP8, validado con la librería **Flake8**. Todo ello ejecutado con **Tox** como *framework* de *testing* para el conjunto.
En cuanto a Node.js, se sigue utilizando -tanto para *testing* unitarios y de integración- el conjunto de **Mocha, Chai, Sinon** e **Instambul**, como *framework*, librerías y validador de cobertura cubierta, respectivamente, y **Plato** como sistema para realizar análisis estáticos del código. Otras herramientas que puede interesar explorar a la hora de complementar el *testing* en Javascript serían: por un lado la librería **Jest**, la cual promete mejoras de rendimiento con respecto a Mocha al ejecutar los test en paralelo, así

como *snapshot testing* para verificar los resultados. Por otro lado, también interesa explorar el *framework* **Stryker,** que permite medir la efectividad de los test con la técnica de *mutation testing*.

_DBs

Como principales BBDD y por categorías comenzando por la *cloud* pública, **AWS Aurora** -dada su escalabilidad como BBDD relacional- se mantiene por delante de otras soluciones, dejando las BBDD tradicionales (MySQL, PostgreSQL, Oracle) para casos de uso fuera del mundo *cloud,* siempre y cuando la propia solución lo requiera.
En el mundo de las BBDD no relacionales, a la ya existente **MongoDB** incorporamos **Neo4J,** como BBDD orientada a grafos y que está dando muy buen resultado en proyectos cuya aplicación es natural.
Cassandra se está convirtiendo en una alternativa muy buena dada su gran escalabilidad, rendimiento y fiabilidad. Por su parte, **ETCD** es una opción muy interesante para BBDD clave/valor pequeñas y distribuidas, por su baja latencia y rendimiento; muy recomendable por su sencillez a la vez que por su potencia.
Por último, **ArangoDB** se mantiene en observación por su capacidad multimodelo, aunque sería justo comparar con sus respectivas alternativas *ad-hoc*.

_Otros

Parece que 2017 ha sido el año por excelencia de los llamados *service mesh,* en parte por la proliferación de arquitecturas orientadas a microservicios. Un *service mesh* se define como una manera sencilla de implementar ciertas buenas prácticas de estas arquitecturas, pero no tanto a nivel de código de nuestro servicio, sino envolviendo al mismo. Esto nos permite no preocuparnos de funcionalidades como *circuit breaking, timeouts, service discovery*... Ejemplos de esta tecnologías pueden ser **Conduit** basado en **Linkerd,** o **Istio** sobre **Envoy**.
También consideramos **Netdata**, una herramienta que permite visualizar y monitorizar métricas varias en tiempo real, centralizadas y recogidas desde un demonio en ejecución en instancias o máquinas distribuidas.
Gracias a la baja latencia que ofrece **gRPC** y al uso que está adquiriendo para soluciones y modelos basados en microservicios, así como el control que aporta con sus interceptores y la facilidad de generar el modelo de mensaje de intercambio o el .proto, lo consideramos como una tecnología a adquirir.

Próximos pasos

_Lenguajes

Continuamos otro año con **Golang**, ampliando su uso y profundizando en más herramientas de su ecosistema, tanto en *frameworks* como en *testing*. En este sentido pasa a ser parte de los lenguajes *core*.
Con **Groovy**, que hasta el momento había pasado desapercibido utilizándose únicamente de forma complementaria para desarrollo de *plugins*, nuestro objetivo es ampliar su uso a otros contextos y aprovecharnos de sus grandes ventajas en ciertos aspectos.

_Microservicios

Dada la gran evolución y necesidad de los microservicios, cobra más importancia el uso de patrones como *Consumer Driven Contracts*, de forma que se puedan testear correctamente estas arquitecturas. Como hemos podido comprobar, **Spring Cloud Contract** cumple este papel y la hemos adoptado este año.
Se integra en Spring una solución que al igual que Zuul, da solución para el enrutado seguro, monitorizado y tolerante a errores. Es **Spring Cloud Gateway**, se explorará esta solución y comprobaremos cómo se integra. También pasaremos a explorar **Spring Cloud Function**, que nos permitirá aprovechar las características de Spring Boot en entornos *serverless*, homogeneizando el funcionamiento de los proveedores o permitiéndonos otro tipo de despliegues.
También consideramos que **Spring Cloud Sleuth** puede facilitarnos el rastreo de nuestras peticiones en las arquitecturas de microservicios.

_APIs

Debido a su madurez, que su uso comienza a estar muy extendido, y que nos ha llamado la atención por su simplicidad, vamos a adoptar **Kong** como API Manager en soluciones que nos permitan aplicarlo.
Se explorará **RSQL,** que tiene la capacidad para realizar filtros complejos en URIs, de forma simple y flexible, especialmente con los operadores lógicos.
A lo largo del año trabajaremos con la combinación **Cognito / API Gateway / Lambda**, puesto que es una solución completa y ágil como arquitectura de API *serverless* en AWS. Nuestro objetivo es explorar esta opción frente a otras que no son puramente *serverless*.

_Mensajes

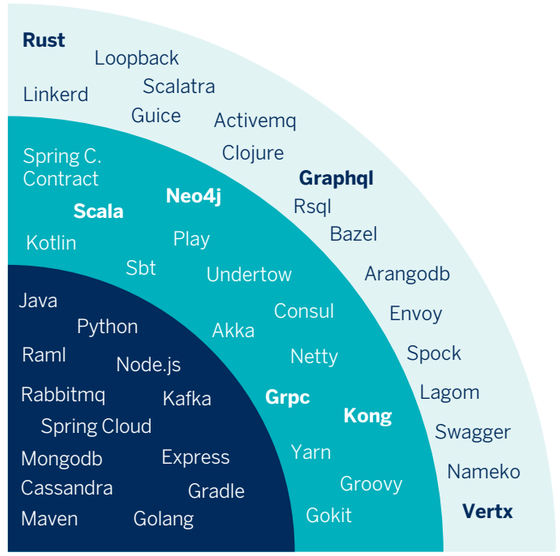
A pesar de que la madurez de **Kafka** es inferior a la de otros sistemas como RabbitMQ, queremos probar su rendimiento en distintos aspectos. Dado que puede ser una mejor opción en algunos casos, especialmente si es interesante reprocesar o conservar más tiempo los mensajes, o en soluciones más orientadas a los productores de mensajes que a los consumidores.
También estamos interesados en estudiar el uso y rendimiento de **ActiveMQ**, y explorar sus puntos fuertes y carencias.

_Framework

Para Node.js, tras tiempo de uso y madurez, se consolida **Express** como el *framework* para este lenguaje. No sin estar atentos a otros posibles *frameworks* que a la larga puedan resultar más interesantes o con ciertas ventajas, como puede ser **Loopback**, el cual analizaremos.
Respecto a Python, hemos adoptado el uso de **Flask** como *framework* principal, debido a su sencillez y potencia, no obstante, seguiremos manteniendo en exploración otros interesantes como son **Django, Bottle y Falcon**, por si mejoran sus prestaciones.

_Servidores de aplicaciones

Por otro lado, **Tomcat** sigue siendo la solución por defecto, por su sencillez de uso.



_Descubrimiento de servicios / Servidores de configuración

En este caso **Consul** sigue siendo la opción por defecto debido a su madurez.

_Gestores de dependencias

Continuamos con los gestores ligados a sus lenguajes de referencia, dados sus buenos resultados: **Maven** para **Java, SBT** para proyectos **Scala, NPM** para proyectos en **Node.js** y **Pip** con ***setuptools*** para **Python**.

_Tests

Como cambios más relevantes, apostamos por **Wiremock** como complemento a **JUnit/Mockito** y **Spock** como alternativa a los anteriores.
También aparecen **Jest** y **Striker**, que nos abren nuevas posibilidades a las ya conocidas en **Node.js**.

_Base de datos

En bases de datos se sigue apostando por las BBDD tradicionales (**PostgreSQL, MySQL, Oracle**) y en el mundo *cloud,* **AWS Aurora**. Donde sí vemos mucho movimiento es en las no-sql. Hay muchas alternativas dependiendo del caso de uso (**Cassandra, ETCD, MongoDB, Neo4J**), de las cuales se realizará un seguimiento, y tenemos el caso de **ArangoDB**, que parece posicionarse como la única multimodelo. ■

RADAR**2018**
Backend

Core Adoptar Explorar

BBVA
Next Technologies

02. Blockchain

Hemos desplegado **Segwit, una importante actualización sobre Bitcoin** que pretende acabar con algunos de los problemas de escalabilidad, al menos, los relativos al throughput

Resumen

Estos últimos años han sido períodos para afianzar la tecnología *blockchain* en cuanto a su percepcion global como un *gamechanger*, pero también han sido años de profundos cambios y choques con la realidad. Hemos visto cómo Bitcoin vivía *fork* tras *fork* por la limitación, ya bien conocida, del número de **transacciones por segundo**. También ha sido el año en el que las **ICOs** han "crecido descontroladamente", para después empezar a verse reguladas en algunos países. También hemos visto cómo empiezan a surgir *blockchains* con sólida base científica, como **Cardano**. Los consorcios privados, especialmente **Hyperledger**, se han afianzado y han surgido nuevas iniciativas, incluso a nivel nacional, como **Alastria**.

_Interacciones con la industria

En BBVA Next Technologies hemos estado al tanto de estas evoluciones. Hemos tomado parte con diferentes actores de la industria para entender sus necesidades y expectativas en relación con la tecnología *blockchain*, y hemos seguido evangelizando y compartiendo conocimiento al respecto.

_Blockchain-Lab

Recientemente hemos arrancado internamente el Blockchain-Lab, iniciativa a través de la cual hemos pretendido coordinar los intereses de los que formamos BBVA Next Technologies, para apoyarnos en nuestro crecimiento dentro de este complejo ecosistema.

Hemos arrancado también internamente el Blockchain-Lab

Los canales para compartir información que hemos puesto disponibles son listas de correo electrónico, canal de Slack y repositorio en Github (dentro de I+D).

_PoCs y pilotos

Recientemente hemos visto en todos los medios la explosión de las ICO, esto es, venta de participaciones virtuales en organizaciones mediante *smart contracts*. La mayoría de estas ICOs han estado basadas en **Ethereum** y se ha realizado una prueba de concepto utilizando la tecnología **Embark**.
También se han comenzado a utilizar soluciones de terceros que tienen una tecnología *blockchain* subyacente. Esto nos permite no sólo conocer estas tecnologías y sus particularidades, si no también validar los casos de uso.

Próximos pasos

Sentadas las bases para coordinarnos, actualmente hay infinidad de temas en los que habrá que mantenerse activo: desde seguir la evolución de los principales actores, analizar las nuevas promesas y probar los *frameworks* que van surgiendo para facilitar el desarrollo.

_Evolución de blockchains ya establecidos

Como comentábamos, el año pasado se ha desplegado **Segwit**, una importante actualización sobre **Bitcoin** que pretende acabar con algunos de los problemas de escalabilidad, al menos, los relativos al *throughput*. Además, también permite desplegar mejoras importantes, como *state channels*, muy útiles para micropagos. De tener éxito, seguramente esto active (aún más) el potente ecosistema alrededor de Bitcoin, incluso con importantes implicaciones en el mundo IoT.
Por su parte, **Ethereum** sigue evolucionando. El año pasado se ha establecido como la primera opción para desarrollar proyectos descentralizados. Al menos, aquellos compatibles con redes públicas. Además, también ha llegado una actualización importante, y es que en colaboración con **Zcash**, han empezado a incluir una funcionalidad para soportar pruebas de conocimiento cero (muy útiles para incluir mejoras de privacidad).
Además, se espera que se produzca el *fork* que active **Constantinople**, la nueva gran *release* de Ethereum, en la que es posible que se migre al

algoritmo de consenso **Casper**, de tipo ***Proof of Stake*** (PoS). Este sería un hito muy relevante, ya que Ethereum sería la primera gran cadena de bloques que migra de *Proof of Work* (PoW) a *Proof of Stake*. Habrá que estar muy pendientes de este proceso, entenderlo y vigilar cómo se ejecuta: habrá muchas lecciones importantes que extraer de él.
Además, con respecto a **Ripple** y su entorno (como Interledger Protocol), aunque el año pasado han hecho menos ruido que los demás, conviene seguir su evolución, ya que están trabajando mano a mano con grandes actores del ecosistema financiero.
En el campo de los consorcios y *blockchains* permisionados, **Hyperledger** se ha posicionado como referente, contando ya con un proyecto "en producción": Fabric. Además, la **Enterprise Ethereum Alliance** y R3 (**Corda**) siguen activos, y en España se ha creado **Alastria**, un consorcio con el fin de unir empresas de distintos sectores para crear su propia infraestructura *blockchain*. Igual que el año pasado, de estas iniciativas pueden surgir importantes adaptaciones que acerquen el mundo de las cadenas de bloques públicas a las corporaciones.

_Resumiendo, próximos pasos:

- Afianzaremos nuestra experiencia con Ethereum, además de seguir de cerca la posibilidad de incluir pruebas de conocimiento cero, y su migración a PoS.
- Afianzaremos nuestra experiencia en Hyperledger Fabric como referente en cadenas de bloques privadas y permisionadas.
- Seguiremos la evolución de Bitcoin, Segwit y los *state channels* de la Lightning Network.
- Seguiremos la evolución de Ripple.
- Seguiremos la evolución de los principales consorcios: Hyperledger, R3, Ethereum Enterprise Alliance y Alastria.

_Blockchains alternativos

Por un lado, se espera que la red principal de **Tezos** empiece a funcionar en breve. Tezos funciona con PoS, e incluye mecanismos para actualizar las normas que lo regulan. Si tiene éxito, puede suponer un empujón importante para los impulsores de mecanismos de consenso más allá del PoW y, por supuesto, sería un paso importante para hacer más ágil la gobernanza de sistemas descentralizados.
Por otro lado, en 2017 se puso en marcha **Cardano**, la primera cadena de bloques con una sólida base científica y con importantes demostraciones de seguridad. Además, funciona con PoS y y ya se ha establecido como una de las principales cadenas alternativas. Actualmente es un referente a seguir, con lo que merece la pena estudiarla a fondo y familiarizarnos con sus procesos y herramientas.

Cardano es actualmente un referente a seguir, con lo que merece la pena estudiarla a fondo

También se ha hecho pública Catalyst, la primera aplicación sobre el sistema **Enigma**, una plataforma para cómputo privado con la ayuda de un *blockchain*. Más allá de la aplicación específica (*trading* de *cryptoassets*), ya interesante de por sí, conviene seguir la pista a la plataforma propiamente dicha, ya que se encuentra en la punta de lanza de las aplicaciones de la cadena de bloques más allá del "traspaso de valor".
Por último, uno de los "blockchains" (estrictamente no lo es) que más ha crecido en capitalización de mercado ha sido **IOTA**, que en lugar de una cadena de bloques, propone un grafo acíclico dirigido (*the tangle*). Las transacciones no se anotan en bloques, sino individualmente, y el requisito para escribir una transacción es realizar una prueba de trabajo consistente en validar dos transacciones preexistentes (vértices). Si bien está orientado a que dispositivos con recursos limitados puedan participar directamente en la red, su diseño presenta aún ciertas incógnitas. La principal es que la estabilidad del sistema asume un volumen de transacciones muy elevado, y hasta que se alcance esa masa crítica, la red depende de un coordinador central.
Por lo tanto, tareas en este aspecto actualmente:

- Explorar Tezos.
- Explorar Cardano.
- Explorar Enigma.
- Seguir la evolución de IOTA.

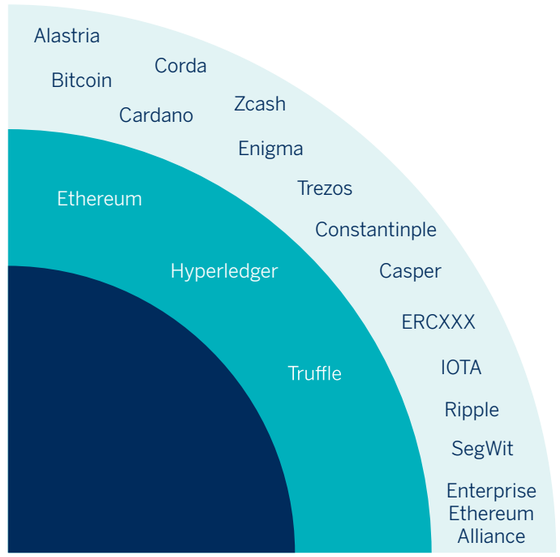
_Herramientas para el desarrollo

Especialmente en el ecosistema de Ethereum, han madurado mucho algunas herramientas para facilitar el desarrollo. Concretamente, **Truffle**, de **Consensys**, se está convirtiendo en una herramienta fundamental que permite definir entornos (desarrollo, pruebas, pre…) y lanzar pruebas unitarias automatizadas, al estilo de los desarrollos tradicionales. También hay otros *frameworks* a los que conviene seguir la pista, como Embark. Viendo su evolución, conviene tener presente actualmente:

- Profundizar en nuestra experiencia con Truffle como *toolset* para desarrollo y pruebas en Ethereum.
- Seguir la pista de otros *frameworks*, como Embark, que ya conocemos.

_Esfuerzos hacia la estandarización

Muchas de las ICOs que han surgido se han adaptado a un estándar de facto que define la forma de crear e intercambiar *tokens* (en este caso, via Ethereum). Este estándar es el **ERC20** (*Ethereum Requests for Comments* número 20). Las propuestas ERC son la forma de gestionar la hoja de ruta del proyecto Ethereum.
Del mismo modo, se están definiendo patrones comunes de interfaces a nivel de aplicación, como los **ERCs 725** y **735**, destinados a la gestión de identidades y atestaciones sobre estas identidades. O el **ERC721**, para activos no fungibles, también útil para otros usos (y que se ha dado a conocer con los *cryptokitties*).
Actualmente está siendo muy interesante monitorizar estos (y otros) esfuerzos de estandarización, y ver cómo se adaptan a casos de usos reales en sus dominios de aplicación.
Se requiere analizar esfuerzos de estandarización ERC20, ERC21, ERC725, ERC735, etc. y aplicarlos en PoCs.



_Más PoCs

Como hemos podido comprobar, el ecosistema sigue siendo extremadamente dinámico, con cambios importantes cada poco tiempo. Por ello, es muy importante que, para no quedarnos atrás, vayamos entendiendo y probando las nuevas contribuciones.
Consecuentemente, actualmente estamos explorando el desarrollo de aplicaciones distribuidas (**Dapps**), sus mejores prácticas e implicaciones del cambio de paradigma. Este tipo de aplicaciones son las compuestas mediante *smart contracts*.
También vamos a realizar PoCs con las *blockchains* mencionadas anteriormente, tanto con las "nuevas" (como Cardano y Tezos) como con las ya existentes, para probar los nuevos cambios que se van introduciendo.
Así, en 2018 seguiremos haciendo PoCs y pilotos para entender la tecnología y sus limitaciones. ■

BBVA
Next Technologies

03. Front

Los últimos años han sido **más estables** que los anteriores **en el mundo del front**

Estado actual

Los últimos años han sido más estables que los anteriores en el mundo del ***front***. VueJS ha sufrido un crecimiento sin precedentes y el resto de actores principales (Angular, React, Polymer y Ember) han seguido evolucionando su producto de una manera menos disruptiva que años anteriores.

Ya podemos sacar conclusiones tras haber testado las anteriores tecnologías más allá de una primera implantación

Por tanto, ya podemos sacar conclusione tras haber testado las anteriores tecnologías más allá de una primera implantación, pudiendo analizar su evolución y mantenimiento en un periodo más extenso de tiempo.
Repasamos para cada ***framework*** los principales hitos acontecidos recientemente:

- **Polymer** y un rico ecosistema generado como parte de un proyecto mayoritario sigue siendo la principal punta de lanza en los desarrollos ***frontend***. Las últimas tendencias de Polymer manejan la posibilidad de integrarse con Redux u otras librerías de gestión de estado, en aras de lograr una base común receptiva a componentes web desarrollados en cualquier librería.
- **React**, tReact, tras cierta inquetud por sus cambios de licencia, está siendo utilizado, junto con **Redux**, en la consola de Ether, en un proyecto relevante de uno de nuestros clientes, dando muestra de su potencial en proyectos que requieren de un rendimiento superior. Se ha detectado el uso mayoritario de **Redux** en otros proyectos de cierta envergadura, ya que otorga la flexibilidad suficiente a cambio de un coste aceptable de complejidad añadida. En este aspecto resulta particularmente relevante escoger el patrón de proyecto adecuado para maximizar la adecuación de la solución de la gestión de estados, manteniendo bajo control la complejidad extra.
- **Angular**, ya en su versión 6, sigue siendo una alternativa fiable a la hora de elegir una única tecnología con la que implementar de forma rápida y eficiente una aplicación de ***front***. Por otro lado, AngularJS sigue funcionando en la compañía y Google lo sigue actualizando regularmente, lo que garantiza que, para proyectos existentes, sigue siendo una opción válida, aunque se están evaluando alternativas de migración.

AngularJS sigue funcionando en la compañía y Google lo sigue actualizando regularmente

- No podemos olvidarnos de **Ember**, que pese a una menor repercusión en el mundo del javascript, está demostrando su solidez y arrojando grandes resultados en cuanto a rendimiento y escalabilidad. Destacan las pruebas de concepto para su uso en aplicaciones web progresivas (PWA) y aplicaciones híbridas, así como la carga ***on-demand*** de determinados componentes de la aplicación.
- Por último, **VueJS**, que ha sido el gran protagonista en los últimos años, muestra ser una librería fiable, ligera y de fácil adopción que está siendo probada en numerosas aplicaciones de la compañía. El proyecto es estable y tiene un ***core*** de desarrolladores reconocidos por la comunidad,

siendo muy probable que este año crezca de manera exponencial. De la mano de **Vuex**, permite implementar SPAs de forma rápida y sencilla y que, gracias a Nuxt, pueden ser perfectamente indexables por los buscadores, algo que le hace ser una gran alternativa a la hora de crear páginas web que requieran de un buen posicionamiento SEO.

Más allá del constante vaivén de ***frameworks***, el ecosistema del desarrollo web ha sufrido cambios que repercuten en la mejora de la estabilidad en los entornos de desarrollo y de rendimiento en los dispositivos finales.

En este sentido, debemos mencionar los siguientes proyectos:

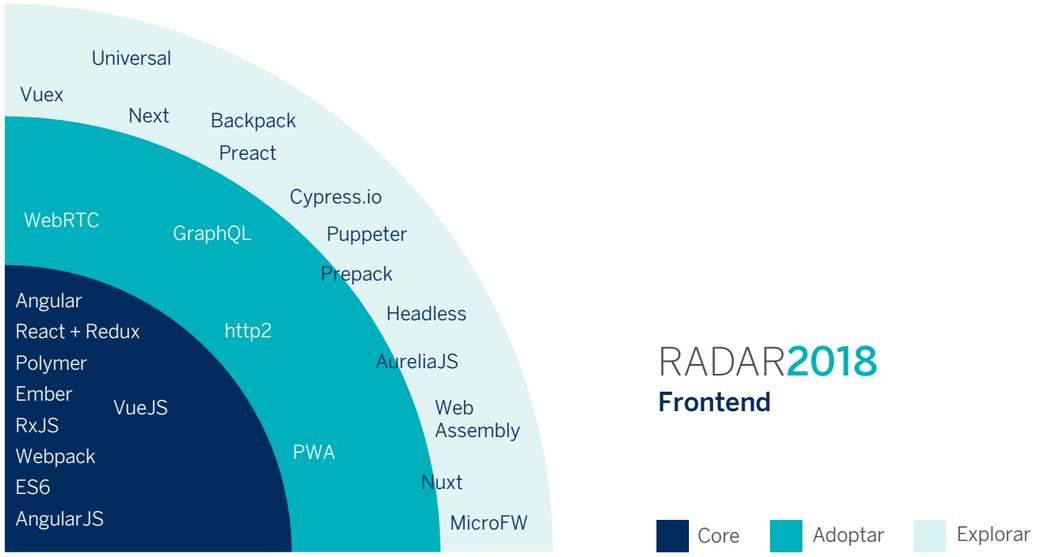
- **RxJS**, la librería con utilidades que facilitan la programación reactiva en Javascript ha llegado para quedarse, siendo pieza fundamental en la gran mayoría de desarrollos de *front*.
- **YARN**, que permite la gestión rápida y estable de dependencias que está siendo utilizada con buenos resultados en varios proyectos.
- **HTTP2**, nuevo protocolo web que resuelve problemas habituales a la hora de servir contenidos de una página web, mejorando sustancialmente el rendimiento.
- **webRTC**, nueva API que permite la comunicación en tiempo real entre navegadores con unas pocas líneas de código.
- Finalmente, se ha generalizado el uso de **CDN** (AWS Cloudfront) y **Firewall** (WAF) en decenas de páginas web corporativas productivizadas por uno de nuestros principales clientes, para evitar ataques (XSS, SQLinjection, ...) y mejorar el rendimiento de las aplicaciones.

Más allá del constante vaivén de frameworks, el ecosistema del desarrollo web ha sufrido cambios que repercuten en la mejora de la estabilidad en los entornos de desarrollo y de rendimiento en los dispositivos finales

Próximos pasos

Con el uso de ES6 totalmente implantado en todas las capas del ***front*** (cliente y servidor), funcionalidades como "***async – await***", ya en las versiones LTS de NodeJS, permiten hacer el código asíncrono mucho más legible y mantenible.

Por otra parte, cada vez más se nota la inclusión de paradigmas funcionales y reactivos en las aplicaciones por su mayor rendimiento. Tanto los proyectos en Angular como en Polymer suelen incluir diferentes arquitecturas de aplicación basadas en Rxjs que son garantes de estabilidad y rendimiento. Con respecto a Aurelia Js estamos testando este ***framework*** y estamos atentos a su evolución, ya que es muy prometedor pero su comunidad e implantación es todavía reducida. Igualmente, han aparecido multitud de librerías y ***microframeworks*** (Moon, Marko, Hyperapp, Quasar Framework, POI, Frint, jsblocks, Svelte/Sapper, Stimulus, Choo) que podrían ser buenas opciones para desarrollos sencillos y sin necesidad de evolución y/o mantenimiento. Finalmente es interesante mencionar los esfuerzos en el ***testing*** automático, en las que las nuevas versiones de ***headless chrome*** prometen integraciones mejores y más simples. ■



BBVA
Next Technologies

04. Movilidad

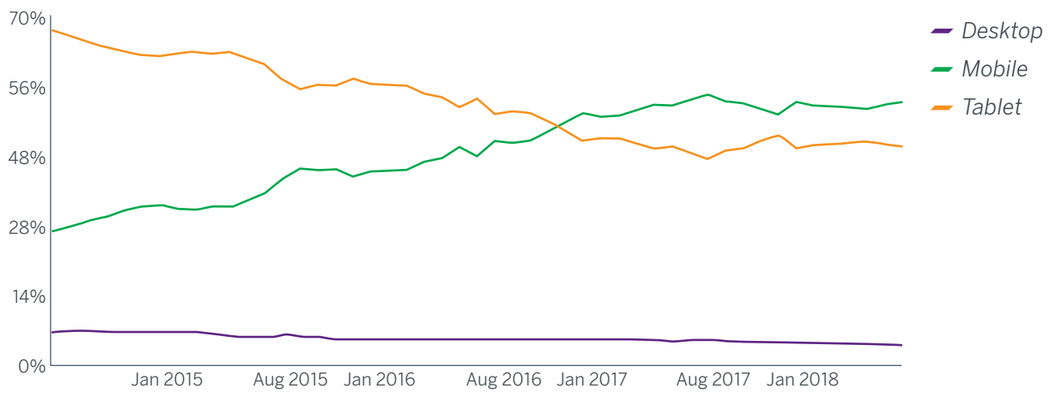
“Mobile first” se consolida como una realidad. El crecimiento de los dispositivos móviles sigue aumentando respecto a los sistemas tradicionales de escritorio

Estado actual

“**Mobile first**” se consolida como una realidad, y es la política que empresas como Google y Apple tienen en cuenta en el diseño de nuevos productos.

La gráfica de esta página muestra cómo el crecimiento de los dispositivos móviles sigue aumentando respecto a los sistemas tradicionales de escritorio.

StatCounter Global Stats
Desktop vs Mobile vs Tablet Market Share Worldwide from Jan 2014 - Jan 2018



Fuchsia, como comentamos el año pasado, es un nuevo sistema operativo que está creando Google para unificar Android junto con Chrome OS.
Aún no se ha presentado nada oficial, pero ya se pueden ver algunas imágenes del propio sistema operativo funcionando. Es algo muy a tener en cuenta en 2018.
Acciones tan cotidianas como pedir comida o enviar dinero a un amigo, ahora pueden realizarse en cuestión de segundos a través de los ***bots***.
Este mercado está creciendo bastante, y aún puede crecer mucho más si Whatsapp integra y da soporte a *bots*. Google Assistant, Siri o Alexa son los asistentes que nos proporcionan las grandes marcas para poder interactuar de una nueva forma con nuestros dispositivos. Estos *bots* requieren de inteligencia artificial y *machine learning* para poder dar respuesta al usuario de una forma coherente. Además la inteligencia artificial está teniendo un crecimiento abrumador, por lo tanto, las aplicaciones móviles deberán adaptarse a este cambio.

Un concepto de aplicación que está teniendo un ligero crecimiento, sobre todo en aplicaciones que no requieren un alto rendimiento, son las aplicaciones híbridas. Estas aplicaciones mezclan desarrollo nativo con desarrollo web. Los *frameworks* más usados son Apache Cordova y React Native.

Acciones tan cotidianas como pedir comida o enviar dinero a un amigo, ahora pueden realizarse en cuestión de segundos a través de los bots

Otro SDK para realizar aplicaciones multiplataforma es **Flutter**, uno de los métodos principales para el desarrollo en Fuchsia.

_Arquitectura y calidad

No por ser parte transversal a la tecnología la arquitectura deja de tener una visión importante aquí. Google ha liberado **Architecture Components**, un conjunto de librerías que ayudan a los desarrolladores a construir aplicaciones más robustas y mantenibles. En todos nuestros proyectos, independientemente de la plataforma y el lenguaje, siempre ponemos foco en la implantación de las arquitecturas que mejor encajen con cada desarrollo.

_Building

Fastlane se ha convertido en la principal *suite* de herramientas *open source* de construcción y despliegue de aplicaciones para dispositivos iOS. El uso de estas herramientas obliga a escribir un conjunto de *scripts* en Ruby, pero la nueva versión de Fastlane permite utilizar Swift, lo que facilita el uso a cualquier programador iOS que tenga conocimiento de este lenguaje. En el mundo Android, **Gradle** continúa siendo la herramienta fundamental para la construcción de aplicaciones, actualmente con soporte para Kotlin como lenguaje de desarrollo, junto con Groovy. **BuddyBuild**, un conjunto de herramientas similar a Fastlane previo pago y con soporte para iOS y Android, aparece en escena por su reciente compra por parte de Apple. El equipo de desarrollo y sus herramientas se integrarán en Xcode, lo que hace entrever un soporte oficial para la integración continua y el despliegue al App Store, por lo que no hay que perder de vista los próximos movimientos, ya que de hacerse realidad las suposiciones, Fastlane pasaría a segundo plano dando paso a estas nuevas herramientas oficiales.

_Languages

En cuanto a lenguajes, **Kotlin** tuvo su mayor impulso en 2017, no solo por su gran crecimiento y aceptación, sino por el soporte oficial de Google. Además de la gran acogida, en el mundo Android es uno de los lenguajes con mayor proyección del momento -¿quizá sea uno de los lenguajes de desarrollo para Fuchsia?-, por su tecnología de compilación a código nativo, que permite la ejecución del código sin una máquina virtual, entre otras funcionalidades.

Kotlin es uno de los lenguajes con mayor proyección del momento por su tecnología de compilación a código nativo

Swift sigue en continua evolución y ya se encuentra en su versión 4.0. Se trata de un lenguaje todavía joven pero con cierta madurez que invita cada vez más a utilizarlo en los nuevos desarrollos. Los cambios producidos en la última versión invitan al programador a dar el salto al paradigma funcional frente al estilo clásico, basado en la programación orientada a objetos.

_Testing

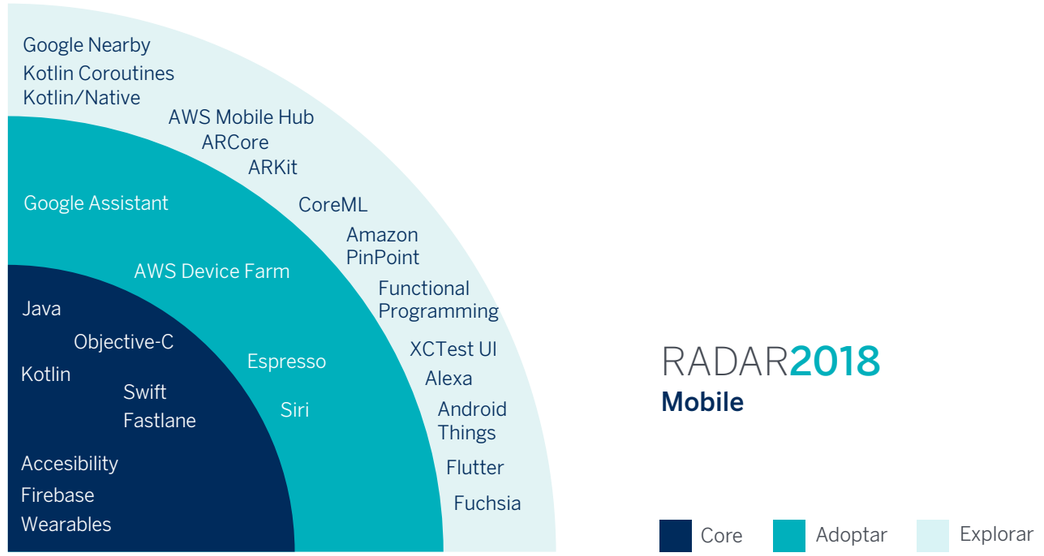
El desarrollo conducido por test (TDD) es una práctica que se está incluyendo en nuestros proyectos para mejorar la calidad de nuestro código. Aplicar ***TDD*** nos ayuda a organizar mejor la forma que tenemos de desarrollar y centrarnos en aspectos concretos de nuestro desarrollo. Para seguir mejorando el desarrollo de nuestras aplicaciones, otra de las medidas que se han llevado a cabo y que debemos seguir investigando es la inclusión de **Sonar** en el ciclo de integración continua.

Próximos pasos

Uno de los campos que más auge están teniendo en los últimos tiempos es la programación reactiva, un paradigma enfocado en el trabajo con flujos de datos finitos o infinitos de manera asíncrona. Es por ello que se van a explorar las posibilidades que ofrecen las implementaciones de **ReactiveX**:

- **RxSwift** para iOS,
- y **RxKotlin/RxAndroid** en el caso de Android.

Por otro lado estamos estudiando algunos *frameworks* de arquitectura como **ArchitectureKit** (en conjunción con **FunctionalKit** dedicado a programación funcional) o **RxFeedBack,** cuya finalidad es simplificar el manejo del estado de una aplicación móvil, facilitar el testeo de la misma, desacoplamiento, etc.
Respecto a persistencia en dispositivos móviles aparece en escena **Realm**, una nueva base de datos enfocada a la programación reactiva que compite con algunas ya conocidas como CoreData o SQLite. Algunas de sus ventajas más notables son: **multiplataforma, eliminación de ORM's**, eliminación de modelos intermedios, y plataforma *online* para servicios de *backend* (**Realm Platform**).
Desde hace ya algún tiempo se habla de las posibilidades que ofrece la realidad aumentada y su inclusión en aplicaciones móviles. Tanto Apple como Google, no ajenas a esto, han presentado sendos *frameworks* de realidad aumentada: **ARKit** y **ARCore** para iOS y Android, respectivamente. No dejaremos pasar la oportunidad de ver qué nos podrían aportar.
Para finalizar, últimamente han aparecido *frameworks* de *machine learning* para dispositivos móviles, una tecnología hasta ahora relegada a aplicaciones de *backend*. En el caso de iOS, Apple ha presentado un *framework* para integrar modelos de *machine learning* pre-entrenados en nuestras aplicaciones y que es capaz entre otras cosas de analizar imágenes, procesar texto, evaluar árboles de decisión, redes neuronales, etc. Además, **Tensorflow**, un *framework* de Google que está disponible para ambas plataformas, parece otra buena opción a tener en cuenta. ■



BBVA
Next Technologies

05. Big Data

El crecimiento del ecosistema de **IoT** ha tenido además **una fuerte influencia sobre las tecnologías big data**, que han ido evolucionando desde modelos puramente batch hacia sistemas de procesamiento en streaming

Estado actual

AWS sigue siendo el lugar donde se han desplegado una gran mayoría de los proyectos de *big data*. Servicios como **S3, Redshift, DynamoDB, Kinesis y EMR** siguen siendo *core* para estas arquitecturas. A este conjunto de servicios se le complementa cada vez más con tecnologías *serverless* como **Lambda y AWS Step Functions**, de cara a realizar tareas de *micro-batching*. Además también se usa **Athena** como complemento para analítica sin aprovisionamiento de cómputo, una tecnología que es muy similar a **BigQuery**.
En cuanto a tecnologías que no se basan en Hadoop, la *suite* compuesta por **Kafka, Cassandra y Spark** sigue siendo la principal alternativa. **Spark** en concreto se ha consolidado como tecnología *core* para procesamiento de datos, mientras que **Kafka** se está convirtiendo en la solución recomendada para hacer análisis sobre datos ingestados en tiempo real.

En cuanto a tecnologías que no se basan en Hadoop, la suite compuesta por Kafka, Cassandra y Spark sigue siendo la principal alternativa

Con respecto a las soluciones para la ingesta de datos, sigue existiendo una gran variabilidad en las tipologías de los datos, aunque se ha observado cierta tendencia al aumento en las ingestas en tiempo real. **Kafka, AWS Kinesis Streams y Azure Events Hub** cubren esta necesidad como plataformas de entrada de datos distribuidas. Además se ha utilizado **Apache NiFi** con buen resultado en procesos de entrada de datos en *streaming* a los que se añaden pequeñas ETL de transformación y enrutado de mensajes. Si bien los volúmenes de carga no han sido masivos, esta herramienta también se ha investigado en el ámbito de IoT para la recolección de datos.

Spark se ha consolidado como tecnología core para procesamiento de datos, y Kafka se está convirtiendo en la solución recomendada para hacer análisis sobre datos ingestados en tiempo real

ElasticSearch se mantiene como tecnología *core* en cuanto a motores de búsqueda, tanto en sus versiones *on-premises* como en los servicios gestionados de **Azure Search y AWS Cloudsearch**.

Próximos pasos

El gran reto actualmente es la entrada en vigor de la GDPR. Las distintas medidas que hay que cumplir en cuanto al tratamiento de datos personales afectan en gran manera a todo este ámbito. Todas las tecnologías que se dan actualmente como probadas y/o *core* se están reevaluando junto a la luz de esta normativa para ver si son compatibles con sus requisitos.
Por otro lado, los distintos proveedores de nube pública ya proporcionan una serie de funcionalidades que dan soporte a esta regulación en cuanto a seguridad y cifrado de datos, pero además han publicado nuevos servicios con el objetivo de dar soporte a su cumplimiento, como **AWS Macie, Microsoft Operations Management Suite en Azure y Google Cloud Security Scanner**.
Tras los buenos resultados obtenidos por **Database Migration Service**, creemos que es conveniente seguir su adopción, probando más integraciones con otras bases de datos, tanto origen como destino, además de usar el patrón CDC cuando sea posible.

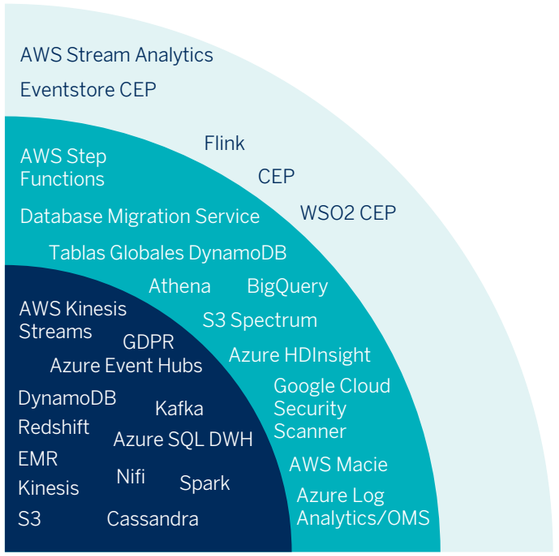
Athena se ha comenzado a usar eventualmente para la realización de consultas puntuales sobre S3, y su utilidad hace que continuemos el proceso de adopción de esta tecnología como parte del core empresarial

De cara al almacenamiento y *data warehousing* sobre AWS, se han lanzado nuevas funcionalidades que queremos testear de cara a la mejora de eficiencia y ahorro de costes sobre procesos. **S3 Select** permite filtrar datos directamente sobre S3, reduciendo carga de trabajo en los procesadores de datos, mientras que **Spectrum** permite a **Redshift** ejecutar consultas directamente sobre **S3** sin haber cargado datos previamente en el *cluster*. La alternativa en Azure es **Azure SQL DWH,** que ha demostrado tener una buena escalabilidad y estabilidad como solución de analítica SQL sobre los datos.

Athena se ha comenzado a usar eventualmente para la realización de consultas puntuales sobre **S3**, y su utilidad hace que continuemos el proceso de adopción de esta tecnología como parte del *core* empresarial.
Recientemente hemos podido ver cómo el crecimiento del ecosistema de IoT ha tenido además una fuerte influencia sobre las tecnologías *big data*, que han ido evolucionando desde modelos puramente *batch* hacia sistemas de procesamiento en *streaming*. Ya son de uso común varias tecnologías de procesamiento en *streaming*, como **AWS Kinesis Analytics, Azure Stream Analytics** en la nube pública o **Spark Streaming** para sistemas independientes de proveedor. Sin embargo, por la propia casuística de los sistemas IoT, se prevé un resurgimiento de los motores de procesamiento de eventos complejos. **Flink CEP, WSO2 CEP y Eventstore CEP** son tres tecnologías que conviene tener en el radar.

Por la propia casuística de los sistemas IoT, se prevé un resurgimiento de los motores de procesamiento de eventos complejos

En cuanto a la visualización de datos, **Azure PowerBI** parece ser la herramienta por excelencia elegida por usuarios de negocio, ya que se integra muy bien con todo el ecosistema de herramientas de oficina, además de ser muy potente. **AWS Quicksight** se posiciona como buena alternativa en entornos de AWS. Pese a contar con una funcionalidad bastante sencilla, su integración con los distintos servicios de su entorno y la potencia del motor en memoria **SPICE** la han convertido en una gran opción nativa de AWS. ■



RADAR2018
BigData

Core Adoptar Explorar

BBVA
Next Technologies

06.
IoT

Diversos fabricantes han lanzado dispositivos, pero **la utilidad de nicho de cada tipo de dispositivo** y el intento de hacerse con el mercado mediante la incompatibilidad, **hace que se avance muy despacio**

La cantidad de productos y servicios ofrecidos alrededor del Internet de las Cosas (**IoT**) ha seguido creciendo, pero aún sigue siendo necesario un gran trabajo para alcanzar la madurez. La heterogeneidad e incompatibilidad de soluciones hace que cada proyecto necesite un gran trabajo de integración. Los dispositivos siguen siendo la pieza que falta. Diversos fabricantes han lanzado dispositivos, pero la utilidad de nicho de cada tipo de dispositivo y el intento de hacerse con el mercado mediante la incompatibilidad, hace que se avance muy despacio.

Estado actual

_Plataformas

En 2017 se lanzó el servicio **Google Cloud IoT**. Aunque su evaluación ha sido en fase ***beta***, nos ha permitido ver las principales diferencias con otras plataformas. La más destacable es la forma de tarificación, utilizando para tarificar la cantidad de datos gestionados, mientras que **Azure IoT** tarifica por lotes de mensajes, y **AWS IoT** tarifica por mensaje, con un límite de tamaño de mensaje.

MQTT ha sido explorado como protocolo y, dado que ha sido la opción elegida por todos los proveedores de servicios ***cloud***, desaparece del radar como tecnología separada.

Sin embargo, la incompatibilidad de los dispositivos con los requisitos de las plataformas está haciendo que en algunos casos no se usen los servicios de IoT. Es decir, que los dispositivos se integren directamente mediante HTTP con la plataforma.

_Pasarelas

El punto de encuentro entre los dispositivos y la nube son las pasarelas. Se han evaluado tres enfoques distintos para pasarelas:

Por un lado se ha evaluado utilizar un ***hardware*** y ***software*** de terceros, mediante el uso de ***hardware*** de Eurotech y la evaluación de las soluciones **Kapua** (***software*** para el dispositivo) y **Kura** (plataforma) y sus versiones propietarias **Everyware Cloud** y **ESF**. Esta es la solución que implementa RedHat para proyectos de IoT. La falta de actividad en las versiones libres nos hizo descartarlas y centrarnos en las versiones propietarias. En las versiones propietarias, la complejidad de desarrollo para la plataforma, usando bibliotecas propias y tecnologías demasiado avanzadas para la funcionalidad, nos hizo descartar continuar con ellas.

El punto de encuentro entre los dispositivos y la nube son las pasarelas

Por otro lado, hemos analizado **AWS GreenGrass**. Esta solución consiste en un ***software*** sobre el que es posible desplegar funcionalidad en un dispositivo autónomo del modo que se realiza en la nube. Sin embargo, un enfoque excesivamente orientado a nube para un dispositivo sobre el terreno hace que sea demasiado dependiente del acceso a Internet para cumplir su cometido. Será necesario seguir su evolución.

Teniendo en cuenta que el ejemplo más extendido de este tipo de ***software*** es el uso de un dispositivo tipo Raspberry Pi, se optó por comprobar la utilidad de una distribución de Linux genérica frente a los ***software*** específicos utilizados. Esta prueba mostró que utilizar herramientas tradicionales para la gestión de sistemas ofrece ventajas sobre algunos de estos ***softwares***, siempre que se tengan ese tipo de perfiles. No obstante, este tipo de ***hardware*** dista mucho de estar preparado para utilidades reales, tanto por capacidades como por resistencia al entorno. Ya que no hemos detectado diferencias en el uso de tecnologías de gestión de sistemas en este ámbito, más allá de tener recursos reducidos, estas salen del mapa de tecnologías del IoT.

Utilizar este tipo de pasarelas nos ha permitido probar **Apache NiFi** dentro de la pasarela. Esta aplicación permite el enrutado y filtrado de mensajes, permitiendo almacenarlos o

lanzar acciones con ellos. Esto se muestra especialmente útil en las pasarelas para que puedan tomar decisiones ***in situ*** en base a los mensajes de sus dispositivos. Sin embargo, se ha mostrado que necesita demasiados recursos para dispositivos de bajas prestaciones, por lo que entra en el radar a explorar Apache ***minifi***.
Otro enfoque utilizado por algunos proveedores para este tipo de ***software*** es la utilización de Yocto Linux en ciertos dispositivos. Esto permite gestionar el ***software*** mediante capas de modificaciones en el sistema. Sin embargo, concluimos que la complejidad de uso no compensaba la funcionalidad ofrecida.
Para la gestión de dispositivos también hemos podido probar **Resin.io.** Desde el punto de vista tecnológico, se trata de un producto muy maduro y estable. Permite trabajar con un flujo de integración continua de aplicaciones de IoT en microcomputadores con arquitectura ARM y desplegar de manera continua sobre contenedores Docker. Además, las funcionalidades de gestión remota y orquestación tienen APIs que permiten automatizar todos los procesos. Sin embargo, lo descartamos por el coste del servicio. El coste de suscripción por dispositivo es excesivo para las prestaciones dadas y difícilmente aplicable a un proyecto con un gran número de dispositivos. Por otro lado, se trata de una tecnología altamente recomendable en el ámbito de la innovación y en el desarrollo ágil de pruebas de concepto y pilotos.

_Comunicaciones y hubs

Recientemente hemos analizado "**LoRa**", normalmente contrapuesto a Sigfox, que ya habíamos analizado en años anteriores. No obstante, LoRa es una tecnología radio más similar a WiFi, con la que es necesario desplegar infraestructura propia. En la actualidad se distribuye principalmente mediante ***kits*** electrónicos y existen pocos dispositivos que lo soporten de serie. Sin embargo, parece haber una tendencia de adopción, por lo que es necesario evaluar su despliegue.
También se han analizado dos de las principales pilas de protocolos de comunicación entre dispositivos: **Zigbee** y **Zwave**. Estos protocolos definen estándares de comunicación en todos los niveles, desde la capa física hasta la capa de aplicación. El problema es la integración de dispositivos de cada uno de los diferentes sistemas, ya que no hay un estándar de interoperabilidad entre ellos. Para solucionar este tipo de problemas existen plataformas como **Samsung Smartthings**. Para resolver la interoperabilidad a nivel de red, **Smartthings** ofrece un ***hub*** que permite conectar dispositivos de múltiples tecnologías (en concreto, soporta tanto Zigbee como Zwave) con su plataforma en la nube. En la plataforma el proveedor ha implementado la integración con dispositivos de otros fabricantes, y es ahí donde se puede conseguir la interoperabilidad a nivel de aplicación, haciendo que el uso de los dispositivos de diferentes tecnologías sea transparente para el usuario. No obstante, esta integración sólo puede realizarla el proveedor con los dispositivos que haya seleccionado.

Próximos pasos

_Plataformas

Los servicios **IoT** de las distintas plataformas, **AWS**, **Azure** y **GCloud** han ido evolucionando y añadiendo funcionalidad, como las necesarias capacidades de **gestión de flotas**, por lo que es necesario adoptar sus servicios actuales y continuar explorando sus capacidades. Es muy importante que empiecen a desplegarse servicios del **plano de gestión** de dispositivos y enfocarlos a que sean meros generadores y consumidores de datos.

_Pasarelas

Es necesario seguir la evolución de **AWS GreenGrass**, ya que se han introducido nuevas funcionalidades, así como evaluar **Azure IoT Edge**. Hasta la fecha, Google ha optado por **Android Things** para este tipo de funcionalidad, aunque mucho más orientado a desarrolladores de aplicaciones móviles que a usuarios de los servicios ***cloud***, por lo que será necesario evaluarlo y determinar las diferencia con sus competidores.

_Comunicaciones

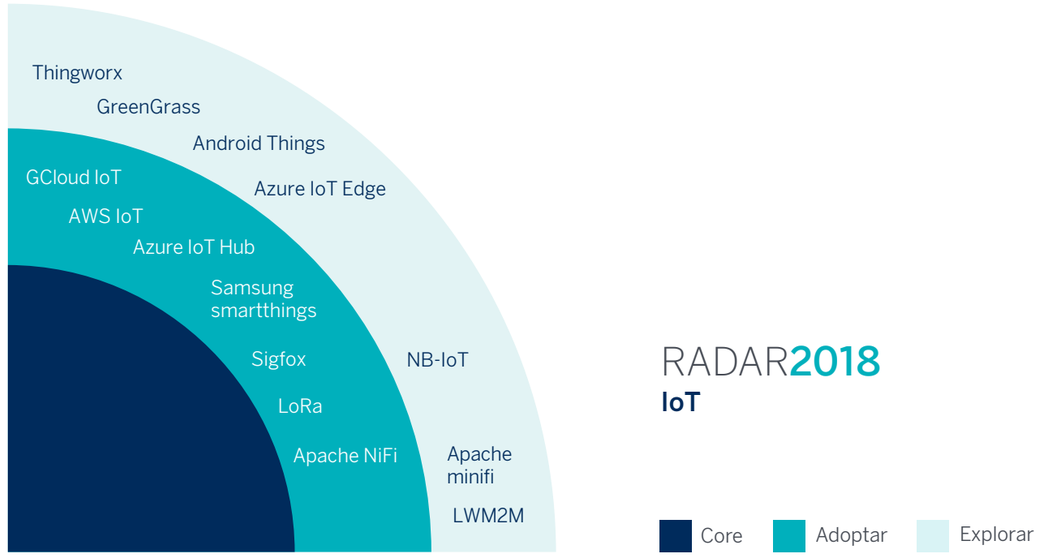
Actualmente está planificado que varios operadores de telecomunicaciones comiencen a ofrecer servicio **NB-IoT** (*NarrowBand IoT*), por lo que es necesario explorar su avance y capacidades. Este tipo de servicios completará el conjunto creado por los ya explorados LoRa y Sigfox.

_Dispositivos

La diversidad y heterogeneidad de los dispositivos actuales hace que sea necesario seguir la evolución del mercado y conocer las tendencias en cuanto a la **integración de dispositivos** en plataformas ***cloud***. Asimismo, es necesario contactar con proveedores de dispositivos con el objetivo de crear un catálogo que nos permita conocer las posibilidades actuales en cuanto a funcionalidad y costes.

_Sistemas confiables

Durante el año pasado, y en previsión de las novedades tecnológicas en años venideros, hemos estado analizando las propuestas tecnológicas para abordar el problema de la seguridad en dispositivos con recursos restringidos. Para ello hemos probado la tecnología de Entornos de Ejecución Seguros (***Trusted Execution Environments*** o **TEEs**) y hemos analizado las diferentes arquitecturas sobre las que se puede implementar.
La tecnología TrustZone de ARM tiene la ventaja de que no requiere de ningún componente ***hardware*** ni ningún ***core*** adicional, y por lo tanto tiene un impacto mínimo en coste y en área. De hecho, ya está presente en la mayoría de dispositivos basados en microcontroladores Cortex-A (la gran mayoría de teléfonos móviles, tablets y dispositivos de *infotainment*). Algunos fabricantes de ***System-on-Chip*** están optando por dar un paso más en el aislamiento ***hardware*** e integran un ***core*** dedicado a operaciones de seguridad crítica, lo que se conoce como enclave (Apple Secure Enclave). La mayor preocupación, de todos modos, es conseguir soluciones de seguridad asumibles por todo tipo de dispositivos IoT, incluso los de coste más reducido, y minimizar el riesgo de ***botnets***.
Aunque el primer microcontrolador basado en la arquitectura ARMv8-M, fue liberado durante 2017, será este año cuando los principales fabricantes saquen al mercado sus primeros chips basados en los diseños de ARM M23 y M33, respectivamente. Estos nuevos modelos vienen bajo la promesa de hacer llegar la seguridad a todos los dispositivos IoT, al añadir la tecnología TrustZone incluso a los dispositivos más restringidos. ■



BBVA
Next Technologies

07.
Cloud 2018

AWS

AWS sigue siendo la cloud pública **donde más proyectos se despliegan en BBVA Next Technologies**

Estado actual

AWS sigue siendo la *cloud* pública donde más proyectos se despliegan en BBVA Next Technologies. Dado que gran parte de los proyectos son proyectos *big data*, servicios como **S3, Redshift, DynamoDB, Kinesis** y **EMR** siguen siendo *core* para estas arquitecturas.
Se ha venido profundizando en tecnologías *serverless* como **Lambda y AWS Step Functions**, de cara a realizar tareas de *micro-batching* y procesos. Se ha comenzado a utilizar **Athena** y **Glue**.
Se han utilizado los servicios tradicionales de *computación* de AWS como son **EC2, ECS y Lambda**. Lambda se emplea de forma consolidada en todos los proyectos, tanto para manejar infraestructura como para despliegue de funciones de computación.
En cuanto a los servicios de almacenamiento, es **S3** el más profusamente usado y el elemento central de casi todas las infraestructuras. Se ha profundizado en el ahorro de costes en este servicio, que puede llegar a ser uno de los más abultados en la factura final.

Respecto a *data warehousing* **Redshift** sigue siendo el servicio central. Además, se ha comenzado a utilizar **Spectrum** para acceso directo a datos de Redshift mapeando datos directamente desde S3 y desde el catálogo de **Glue**.
También se ha explorado el servicio de ***machine learning*** y desarrollado proyectos con el *framework* **Tensor Flow**.

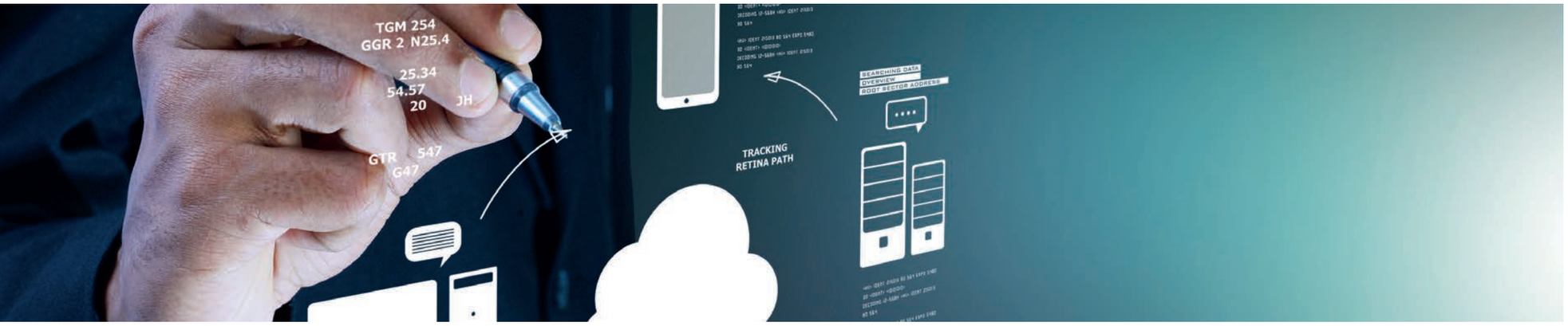
Próximos pasos AWS

AWS ha liberado una gran cantidad de nuevos servicios que se deberán ir incorporando en la medida de las necesidades de los proyectos que se desplieguen.
Para comenzar, respecto a contenedores, tenemos **AWS Fargate**, que es el servicio para el despliegue y administración de *containers* de Docker sin tener que administrar infraestructura. Simplemente basta con subir la imagen del contenedor, especificar los requisitos de recurso y el servicio lanzará los contenedores en segundos.

AWS ha liberado una gran cantidad de nuevos servicios que se deberán ir incorporando en la medida de las necesidades de los proyectos que se desplieguen

Fargate administra todo el escalado e infraestructura necesarios para ejecutar contenedores con alta disponibilidad. No es ya necesario decidir cuándo escalar *clústers* o agruparlos para un uso óptimo. Se puede lanzar cualquier número de contenedores en cuestión de segundos y escalarlos fácilmente para ejecutar aplicación, ya sea en modo *batch* o como servicios web.

AWS Elastic Container Service for Kubernetes (EKS) facilita la ejecución de *clusters* de Kubernetes en AWS de manera gestionada. Esto significa que no hace falta gestionar fallos en nodos o discos: o la red entre los distintos nodos o el *cluster* se aprovisiona y mantiene directamente por el servicio mediante el CLI, SDK o Consola web de AWS.
En el campo de integración y desacoplamiento de aplicaciones aparece el servicio **AWS MQ**. Amazon MQ es un servicio de agente de mensaje administrado para Apache ActiveMQ que facilita configurar y operar intermediarios de mensaje en la nube. Amazon MQ trabaja con aplicaciones y servicios existentes sin la necesidad de administrar, operar o mantener un sistema propio de mensajería.
Respecto a bases de datos, una novedad es **AWS Neptune**, servicio de base de datos orientado a grafos, totalmente administrado en modo SaaS. Neptune permite realizar consultas en los lenguajes de consulta de grafos como Apache TinkerPop, Gremlin y SPARQL.



AWS Aurora ha evolucionado a una configuración ***multicluster*** con esta característica, Aurora permite crear múltiples nodos de escritura, escalar ambas lecturas y escrituras a través de múltiples zonas de disponibilidad e incrementar la escalabilidad y la disponibilidad. La versión preliminar estará disponible para la edición de Amazon Aurora compatible con MySQL. Además, se liberará la versión AWS Aurora Serverless. Este nuevo modo ***serverless*** ahorrará tiempo y dinero al ajustar automáticamente la capacidad de la base de datos para que se correspondan con las necesidades de la aplicación. La versión preliminar estará disponible también para la edición de Amazon Aurora compatible con MySQL.

Aurora permite crear múltiples nodos de escritura, escalar ambas lecturas y escrituras a través de múltiples zonas de disponibilidad e incrementar la escalabilidad y la disponibilidad

AWS DynamoDB ofrecerá **tablas globales**. Las tablas globales se construyen sobre la huella global de DynamoBD para ofrecerle una base de datos completamente administrada de regiones y másteres múltiples que le ofrecen un rendimiento de lectura y escritura global para las aplicaciones de escala masiva con usuarios dispersos globalmente. Las tablas globales manejan el trabajo difícil de la replicación automáticamente de datos entre las regiones y resuelven los conflictos de actualización, al permitir que los desarrolladores se enfoquen en la lógica de la aplicación cuando construyen aplicaciones distribuidas globalmente. Es una característica semejante, en cuanto a disponibilidad global, al almacén de datos Table Storage de Azure.
Respecto al cumplimiento de la GDPR y securización, se liberó el nuevo servicio **AWS GuardDuty. GuardDuty** es un servicio de detección de amenazas que utiliza los *logs* de Cloudtrail y VPC Flow *logs* para detectar anomalías de seguridad, como comunicación con nodos de redes Tor, escaneo de puertos, cambios en políticas de *password*, etc. Permite enviar eventos a CloudWatch y realizar seguimiento de los riesgos en función de su criticidad. Los informes de vulnerabilidades de GuardDuty de diferentes cuentas se pueden centralizar en una cuenta maestra para facilitar su supervisión.

AWS Macie es un servicio para descubrir, clasificar y proteger datos sensibles en AWS. El servicio monitoriza en tiempo real los *logs* de CloudTrail y también puede integrarse con fuentes de datos como Amazon S3 (la única disponible actualmente). Macie detecta información sensible, como datos personales, y proporciona cuadros de mando, informes de riesgo y alertas relacionados con el uso que se está haciendo de la misma.

Una de las funcionalidades más atractivas de AWS Macie es la capacidad para detectar brechas de datos, requisito fundamental para el cumplimento de la GDPR

Una de las funcionalidades más atractivas del servicio es la capacidad para detectar brechas de datos, requisito fundamental para el cumplimento de la GDPR. Al igual que GuardDuty, los informes de riesgo de Macie se pueden centralizar en una cuenta maestra que agregue los resultados de varias cuentas hijas.

Respecto a *machine learning* y aprendizaje automático, **AWS SageMaker** es un servicio gestionado que permite desarrollar y productivizar modelos de *machine learning* de forma sencilla. El desarrollo, entrenamiento, *testing* y despliegue se puede realizar tanto de forma programática contra la API del servicio como a través de *notebooks* de Jupyter.
Si hablamos de IoT, AWS ha introducido gran cantidad de nuevos servicios y capacidades.
Se ha liberado **IoT Device Management**, que ofrece capacidades de administración de dispositivo que facilitan incorporar, organizar, monitorear y administrar remotamente los dispositivos IoT a escala mediante su ciclo de vida. **IoT Analytics** es un servicio de análisis de IoT completamente administrado que recolecta, procesa, enriquece, almacena y analiza los datos del dispositivo de IoT a escala. **AWS IoT 1-Click** es un servicio que facilita que los dispositivos simples activen las funciones lambda de AWS que ejecutan una acción específica. Con AWS IoT 1-Click se puede elegir la acción para su dispositivo al seleccionar una de las funciones de AWS Lambda predefinidas. Por su parte, **AWS IoT Device Defender** es un servicio completamente administrado que lo ayuda a asegurar su flota de dispositivos de IoT. **Device**

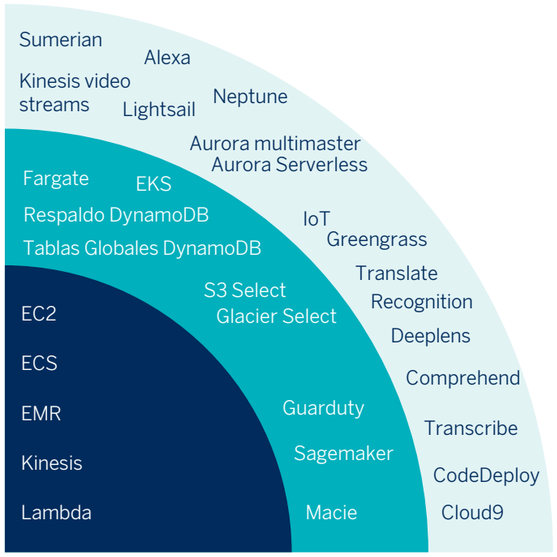
Defender audita continuamente las políticas de seguridad asociadas con sus dispositivos para asegurarle que no se están desviando de las prácticas de seguridad.
Amazon FreeRTOS es un sistema operativo de IoT para los microcontroladores que hacen que los dispositivos pequeños y de bajo consumo sean fáciles de programar, implementar, asegurar, conectar y mantener.
Amazon FreeRTOS ofrece un sistema operativo central como bibliotecas de *software* que facilitan programar los dispositivos basados en microcontroladores conectados y recolectan datos desde ellos para las aplicaciones de IoT.
AWS IoT Core agrega autorizadores personalizados y ahora puede vender las credenciales de AWS para los dispositivos.
A esto hay que añadir las nuevas características de AWS Greengrass. Primero, las funciones lambda de AWS que se ejecutan en los dispositivos **Greengrass Core** pueden interactuar nativamente con las capacidades del dispositivo de alojamiento subyacente. Segundo, Greengrass ahora también puede usar el conocido protocolo de mensaje industrial OPC-UA. Tercero, ahora puede actualizar remotamente el *software* Greengrass Core para aprovechar las nuevas características, las correcciones de errores y las mejoras de seguridad.

También, respecto a Greengrass, AWS pondrá a andar la inferencia de ML de AWS Greengrass que permite implementar y ejecutar una inferencia de ML de manera local en los dispositivos conectados. Al hacer inferencia en los dispositivos conectados se reduce la latencia y el coste de enviar los datos del dispositivo a la nube para hacer una predicción. En lugar de enviar todos los datos a la nube para la inferencia de ML, la inferencia de ML se realiza justo en los dispositivos, mientras que los datos se envían a la nube solamente cuando necesitan más procesamiento.
Respecto al aprendizaje automático surgen servicios interesantes e innovadores que abren nuevas posibilidades. **AWS DeepLens** es una cámara de video inalámbrica habilitada que empareja el kit desarrollador de cámara HD con un conjunto de proyectos de muestra para ayudar a los desarrolladores a aprender los conceptos de aprendizaje de la máquina.
Amazon Comprehend es un servicio de procesamiento de lenguaje natural (NLP) que usa el aprendizaje de máquina para encontrar comprensiones y relaciones en el texto.
Amazon Rekognition Video es un servicio de análisis de video impulsado por el aprendizaje profundo que rastrea personas, detecta actividades y reconoce objetos, celebridades y contenido inadecuado.

Respecto al aprendizaje automático surgen servicios interesantes e innovadores que abren nuevas posibilidades

Amazon SageMaker es un servicio administrado completamente que permite a los científicos y desarrolladores de datos que construyan, capaciten e implementen rápida y fácilmente modelos de aprendizaje a escala. **Amazon Translate** es un servicio de traducción de máquina neural que ofrece traducción de idiomas accesible, de alta calidad y rápida. **Amazon Transcribe** es un servicio de reconocimiento de discurso automático (ASR) que facilita a los desarrolladores agregar capacidades de voz a texto a sus aplicaciones.
Si hablamos de evoluciones en conectividad, redes y entrega de contenido, se debe tener en cuenta que se debería usar en todos los proyectos que se emprendan los nuevos servicios y *features* al respecto.

AWS PrivateLink se utiliza para acceder a las aplicaciones SaaS de terceros desde su Virtual Private Cloud (VPC) sin exponer su VPC a la internet pública y para conectarse a servicios a través de cuentas y VPC diferentes dentro de sus propias organizaciones, lo que simplifica significativamente su arquitectura de red interna. Además serán posibles las **interconexiones de VPC entre regiones**.
También será posible **integraciones de VPC privadas de Amazon API Gateway**, de forma que se puede proveer acceso a recursos HTTP(S) dentro de una Amazon Virtual Private Cloud (VPC) sin exponerlos directamente a la internet pública.



BBVA
Next Technologies

07.
Cloud 2018

Microsoft Azure

Microsoft Azure sigue siendo el **segundo proveedor** de nube pública **a nivel mundial** y sigue recortando terreno con AWS

Estado actual

Microsoft Azure sigue siendo el segundo proveedor de nube pública a nivel mundial y sigue recortando terreno con AWS. La expansión global sigue acelerándose y actualmente cuenta con 50 regiones en todo el mundo. Se han abierto centros de datos en India, Australia, Corea del Sur, China, Japón, UK, Francia, Alemania y se han anunciado próximas aperturas en Suiza, Emiratos Árabes Unidos y Sudáfrica.
La oferta de servicios en Microsoft Azure sigue enriqueciéndose a gran velocidad y se están presentando nuevos servicios que ofrecen funcionalidad no presente en otras nubes públicas, como por ejemplo el servicio gestionado de Kubernetes **AKS**.
Microsoft Azure sigue apostando por la innovación y la adopción de tecnologías **Open Source**. Por poner un ejemplo, actualmente más del 50% de los servidores que se despliegan en los centros de datos de Azure son Linux.
En el caso de *big data*, Microsoft ha decidido prescindir de Windows Server en el despliegue de tecnologías Hadoop y, por ejemplo, tecnologías como Spark, Kafka o Storm se despliegan sólo sobre Linux dentro de la oferta gestionada de **HDInsight**.
A nivel de lenguajes, aparte de .NET, observamos que están ganando relevancia lenguajes como Node.js, Python, R y SQL.

En BBVA Next Technologies, en los últimos años hemos adquirido experiencia con Azure SQL Data Warehouse, Power BI y algunos de los Cognitive Services

En BBVA Next Technologies, en los últimos años hemos adquirido experiencia con Azure SQL Data Warehouse, Power BI y algunos de los Cognitive Services.
Observamos que Azure **SQL Data Warehouse**, la base de datos MPP de Azure, se posiciona como una de las piezas centrales en muchos de los proyectos que requieren hacer BI (*Business Intelligence*) sobre una gran cantidad de datos.
Power BI es una herramienta muy demandada por los clientes y sobre la que deberíamos adquirir un mayor conocimiento. De hecho, muchas veces, en los proyectos, los clientes piden que la solución incluya Power BI independientemente del ***cloud*** donde se despliegue el resto de la solución.
También hemos utilizado por primera vez el servicio **Visual Studio Team Services** en algunos proyectos, para la gestión parcial o completa del ciclo de vida del ***software,*** con muy buenos resultados.
Por otra parte, **Cosmos DB** ha pasado a ser un servicio ***core*** dentro de Azure, con lo que está presente en todos sus centros de datos.
Las bases de datos Cosmos DB son bases de datos NoSQL distribuidas globalmente y con soporte para diferentes tipos de modelo y APIs. Por ejemplo, soporta clave/valor, Mongo DB, Document DB o Gremlin y muy recientemente se ha añadido soporte para Cassandra.
En cuanto a bases de datos SQL, Azure ha añadido como servicios gestionados, con todos los beneficios del ***cloud***, las bases de datos MySQL, MariaDB y PostgreSQL.
A nivel de ***partnership*** con Microsoft, en BBVA Next Technologies conseguimos el estatus **Gold Partner** en 2017 (veníamos de ser Silver Partner el año anterior). También el año pasado hemos conseguido las competencias Cloud Platform, Data Analytics y Data Platform y hemos dado un empujón fuerte al número de personas certificadas en tecnologías Azure, pues actualmente contamos con más de 10 personas certificadas y más de 20 certificaciones.

Próximos pasos

Microsoft Azure sigue innovando a gran velocidad y recomendamos ir incorporando en los proyectos los servicios de Azure que se encuentren en General Availability y que hayamos probado y validado en BBVA Next Technologies. Por otra parte hay una gran cantidad de servicios interesantes en Preview que deberíamos explorar con pruebas de concepto. A continuación, indicamos algunas de las tecnologías que nos parecen más interesantes.

_Confidential Computing (Early Adopters)

Tecnología aún en desarrollo en Microsoft y sobre la que debemos hacer un seguimiento por sus implicaciones futuras.
El objetivo de esta tecnología es garantizar que el código que se ejecuta no ha sido modificado por algún agente malicioso e impedir que nadie pueda acceder a los datos en claro una vez estén siendo procesados. Los datos sensibles se encuentran siempre encriptados fuera del entorno de ejecución protegido. Con esta tecnología, ningún proceso o usuario con permisos de administrador en la máquina o incluso el propio Microsoft podría acceder a los datos una vez están siendo procesados.

Confidential Computing permitirá subir al cloud público cargas de trabajo con requisitos de seguridad estrictos

Esta tecnología permitirá subir al ***cloud*** público cargas de trabajo con requisitos de seguridad estrictos, como por ejemplo soluciones que trabajen con información de salud de las personas, con información financiera, etc.

_AKS

Servicio gestionado de Kubernetes en Azure que facilita la gestión de los ***clusters***, al no tener que preocuparnos por la capa de control. El servicio ofrece toda la funcionalidad de Kubernetes y las ventajas de ser un servicio gestionado en nube. Introduce controles de ***autohealing, autoupgrade***, facilidades de escalado, integración de la monitorización con Azure Monitor, etc.
Crear un ***cluster*** gestionado de Kubernetes es tan sencillo como ejecutar un comando CLI de una única línea. Este servicio aporta grandes beneficios, sobre todo en prácticas DevOps, y está teniendo una excelente aceptación entre los clientes.

_HDInsight

Servicio gestionado de Azure, esencial en soluciones de ***big data*** donde se requiere procesar grandes volúmenes de información de forma rápida, sencilla y eficiente en costes.
HDInsight es la distribución en Azure de los componentes Hortonworks Data Platform (HDP). Incluye despliegue de *clusters* Hadoop, Spark, Kafka, Storm, HBase, R, etc.

_Data Lake Analytics

Servicio de ejecución de trabajos de analítica sobre volúmenes enormes de datos. Es un servicio completamente gestionado, donde no es necesario gestionar ninguna infraestructura. Solo hay que definir los trabajos a realizar e indicar el número de nodos de procesamiento.
Los trabajos se programan en U-SQL. Es un lenguaje de tipo SQL que se integra con los lenguajes C#, Python o R, para poder implementar lógica compleja. El servicio descompone automáticamente el trabajo en tareas y las distribuye en tantos nodos de computación como se indique. Se pueden tener cientos de nodos trabajando en paralelo.

_Data Factory

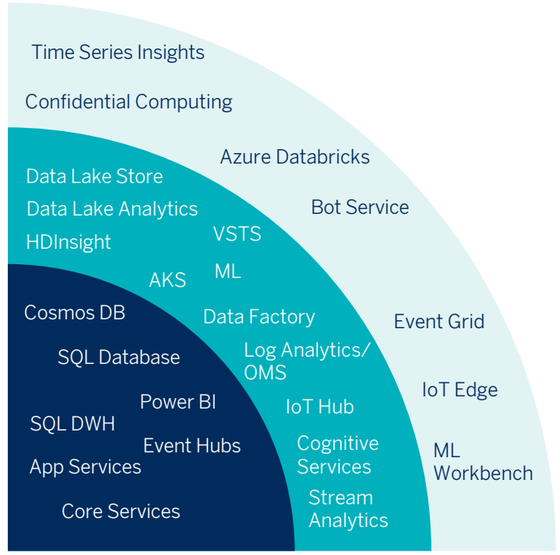
Es uno de los servicios más útiles de Azure, necesario en cualquier proyecto donde se necesite trabajar con datos. En este servicio se define, orquesta y programa la ejecución de trabajos de integración, transformación y movimiento de datos de tipo ETL/ELT.
Está disponible la versión dos del servicio. El nuevo modelo de ***pipeline*** permite construir flujos de control complejos de forma visual en el portal web de Azure.
Data Factory incluye una gran cantidad de conectores con todo tipo de fuentes de datos externas. También se ha incluido la capacidad de ejecutar paquetes SSIS.

_Azure Databricks

Azure y Databricks han llegado a un acuerdo para integrar la plataforma Databricks en Azure como un servicio más. Databricks es una herramienta muy interesante para el análisis de datos o construcción de modelos de *machine learning*. Es una plataforma gestionada que permite trabajar con ***clusters*** de Spark de forma muy sencilla. Utiliza ***notebooks*** como herramienta de desarrollo y permite su ejecución como trabajos de Spark.
Los equipos de trabajo pueden colaborar definiendo *workspaces* y se integra la seguridad con Azure Active Directory.

_Azure IoT Edge

Conjunto de capacidades que permiten acercar el *cloud* a los dispositivos. Por un lado se conecta con dispositivos IoT, que pueden o no tener conectividad a Internet, y por otro lado con el servicio Azure IoT Hub.
Cuenta con una gran cantidad de conectores con diferentes tipos de dispositivos y permite el despliegue de servicios ***cloud*** como modelos de Azure ML, Functions o Cognitive Services.
Por ejemplo, se puede entrenar un modelo de **ML** en el ***cloud***, con los datos que recibe de los dispositivos, y una vez listo, desplegarlo mediante contenedores en Azure IoT Edge, para que responda inmediatamente ante problemas que puedan surgir en los dispositivos.



_Azure Bot Service

Este servicio proporciona todo lo necesario para construir, desplegar, probar, monitorizar y administrar *bots*. Los *bots* se pueden desplegar en App Service en modo de reserva de capacidad o en modo ***serverless,*** donde el coste viene dado por el número de ejecuciones y recursos consumidos por el *bot*.
Los *bots* se integran con gran cantidad de canales como Telegram, Slack, Facebook, Skype, etc. Se integran con métodos de pago y permiten el uso de tarjetas adaptativas. También se integran con servicios de voz, con servicios de reconocimiento de intenciones a partir de texto, etc.

_Machine Learning Workbench

Nuevas herramientas y servicios para los científicos de datos, que aceleran los desarrollos de modelos de ML. Incluyen algoritmos AI que facilitan los trabajos de limpieza de datos, infiriendo reglas a partir de muestras.
Permiten productivizar los modelos de ML en contenedores, con lo que podemos desplegarlos en cualquier sitio, como por ejemplo, en cualquier nube pública, en la máquina de desarrollo del científico de datos, en teléfonos móviles, etc. ■

RADAR**2018**
MICROSOFT AZURE

Core Adoptar Explorar

BBVA
Next Technologies

07.
Cloud 2018

Google Cloud Platform

Google Cloud ha hecho énfasis en ofrecer servicios de transformación de datos, desde la capa de ingesta hasta la capa de presentación, pasando por el procesamiento de datos

Estado actual GCP

Google Cloud ha intentado reforzar su presencia en la nube basándose en dos enfoques principales:

- **Servicios de *machine learning*** mediante el uso de API de modelado de lenguaje natural, vídeo, imágenes o texto. Google ha realizado una apuesta decidida por las APIs de manipulación de datos y ***machine learning***. Así ha puesto disponible un servicio de **Cloud ML** basado en el ***framework*** de referencia Tensorflow, que permite realizar modelos y escalar de manera sencilla sin necesidad de realizar una configuración compleja. Además, ha potenciado y mejorado su servicio de **APIs**, algunas de ellas ampliamente usadas por la comunidad, como el servicio de **Google Translate** o el **API de Vision**.
- Orquestación de infraestructura a través de **Kubernetes**. Otro de los servicios destacados de Google es su servicio de orquestación y administración de contenedores en ***cluster*** **Google Kubernetes Engine**, anteriormente, Google Container Engine. Dicho servicio ha cogido fuerza, pudiendo aprovisionar imágenes Docker a través de un servicio de registro de contenedores propio como Google Container Engine y simplificando todavía más la configuración necesaria para desplegar y administrar un ***cluster*** de máquina dentro de la nube de Google. También integra una herramienta como Stackdriver para el registro y supervisión del propio ***cluster***.

Google Cloud ha intentado reforzar su presencia en la nube basándose en dos enfoques principales

Además de estos dos enfoques, Google Cloud ha hecho énfasis en ofrecer servicios de transformación de datos, desde la capa de ingesta hasta la capa de presentación, pasando por el procesamiento de datos. En este sentido alguno de los servicios que están ahora disponibles, bien como servicio *beta* o 100% productivo, son:

- **Google Dataflow**. Servicio de procesamiento tanto en ***batch*** como ***streaming*** fundamentado en el proyecto Apache Beam. Google ha diferenciado el ***framework*** de procesamiento de ***streams Beam*** del servicio de procesamiento en sí, liberándolo como proyecto de Apache para la comunidad con el objetivo de que se convierta en el ***framework*** de referencia de procesamiento de ***streams***. En ese sentido, librerías como Flink o Apache Spark están adoptando el modelo de procesamiento definido por Beam. Especialmente relevante y significativo del éxito de Beam es que Apache Spark haya construido su módulo Structured Streaming bajo el paradigma definido por Beam. Dataflow en Google Cloud existe como motor de procesamiento sobre Apache Beam. Una de las características de Google DataFlow es la autoescalabilidad en función de la carga (se apoya sobre Apache Beam, APIs para Java y Python). Además de como motor de procesamiento, Dataflow se está convirtiendo en el "pegamento" que permite unir distintos servicios de la plataforma de Google a través de la manipulación de los datos.
- **Google DataPrep**. Servicio de tratamiento de datos de manera visual que permite realizar limpiado y manipulación de datos. No es necesario para el científico de datos operar ni desplegar ningún servicio, sino simplemente acceder a la interfaz visual (o a través del API). Las manipulaciones realizadas sobre los datos serán ejecutadas por DataFlow de manera autoescalable,

no teniéndose que preocupar así por la infraestructura. DataPrep está cogiendo bastante tracción en la comunidad, ya que permite realizar transformaciones de complejidad media de manera muy intuitiva, cubriendo el hueco existente entre soluciones programables más generales como ***notebooks*** y manipulación de datos directamente con macros de Excel, por ejemplo. Además, es compatible con bastante número de servicios de Google, lo cual lo hace todavía más usable.

- **Cloud IoT Core**. Aunque es un servicio que todavía tiene que madurar, el hecho de que Google ofrezca una plataforma tan potente como Android Things puede ser un hecho diferenciador respecto a la competencia una vez que la plataforma ofrezca toda la potencia que tiene.
- **Google Data Studio**. Te permite construir informes de una manera muy sencilla sobre un gran número de fuentes orígenes como Big Query, Google o Youtube Analytics, Adwords, Cloud Storage... Es un servicio que no ofrece actualmente la potencia de manipulación y de consulta que pueden ofrecer sus competidores, pero por otro lado se trata de la más sencilla de usar, es gratuita y, como se ha comentado, integra un gran número de fuentes. Se espera que salga de modo *beta* para convertirse en un proyecto principal.

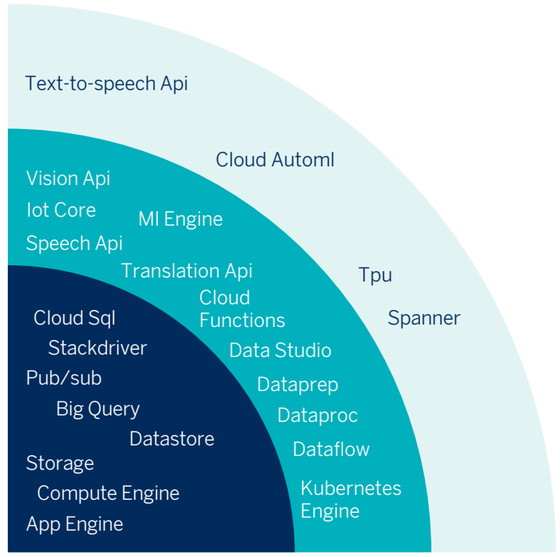
En todo resumen que se realice de Google Cloud siempre es necesario comentar **BigQuery**, probablemente el servicio estrella de la plataforma de Google. Además de la mejora continua del motor de procesamiento Dremel sobre el que se basa BigQuery, se ha trabajado en la integración con distintos servicios manteniendo la separación entre la capa de almacenamiento y procesamiento que otros competidores han ido adoptando en años anteriores.

Próximos pasos

Probablemente el servicio más ilusionante actualmente en la plataforma de Google sea **Cloud AutoML,** que permitirá entrenar modelos propios de *machine learning* de manera rápida y sencilla. El primer servicio disponible es el API de Vision. Es un proyecto todavía en estado ***alpha*** que se espera adquiera madurez durante este año.

A nivel de infraestructura se espera que **Google Kubernetes Engine** siga incluyendo nuevas características sobre el servicio. Además está en *beta* el servicio Cloud Composer, basado en Airflow, que permite gestionar ***pipeline*** de ejecución en la nube.

En la plataforma de Google, Cloud Auto ML permitirá entrenar modelos propios de machine learning de manera rápida y sencilla



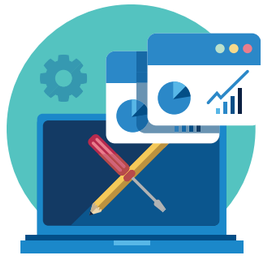
RADAR2018
GCP

Core Adoptar Explorar

A nivel de procesamiento se prevé que **Dataflow** agregue compatibilidad con más servicios de Google, convirtiéndose más todavía en el enlace entre la capa de ingesta y explotación de los datos en Google Cloud. En el caso de **Dataprep** se espera que en este año siga cogiendo fuerza. Además se esperan mayores integraciones con **Google Cloud IoT**. ■

BBVA
Next Technologies

08. DevOps



No todo son las herramientas, ni la automatización. **BBVA Next Technologies** ha hecho un esfuerzo para **extender la cultura DevOps** a sus procesos y a la comunicación entre las personas

Estado actual

Si se analiza el número de tecnologías involucradas desde el momento de la concepción de un proyecto hasta su puesta en producción se comprueba que la cadena de herramientas necesarias en años anteriores sigue aumentando como resultado de la aparición de nuevos retos. Sin embargo, empieza a vislumbrarse un estándar de facto en el mercado en cuanto a las herramientas a utilizar en el *pipeline* de producto ***software (Plan - Code - Build - Test - Deploy - Run - Monitor - Manage - Notify).***

Si bien se ha superado la fase en la que se plantea operar una aplicación de forma manual con *releases* poco frecuentes y se trabaja a diario con tecnologías básicas orientadas a automatización de tareas, han aparecido nuevos retos propios de una fase donde es necesario escalar las soluciones. Esta fase requiere seguir invirtiendo esfuerzos en DevOps, tanto a nivel tecnológico como cultural, retener a aquellos perfiles más experimentados y trabajar en formar o atraer otros perfiles que encajen en este nuevo escenario. Además, la apificación de las herramientas de DevTools para conseguir atender como servicio peticiones comunes a las herramientas se vuelve una prioridad de cara a la integración.

La apificación de las herramientas de DevTools para conseguir atender como servicio peticiones comunes a las herramientas se vuelve una prioridad de cara a la integración

Tras la generación de una masa crítica de perfiles con conocimientos sobre DevOps durante años anteriores gracias a la cantidad de formación disponible, es momento de perfeccionar y extender DevOps al resto de la organización.

No todo son las herramientas, ni la automatización. BBVA Next Technologies ha hecho un esfuerzo para extender la **cultura** DevOps a sus procesos y a la comunicación entre las personas.

En este sentido es clave seguir construyendo equipos alrededor del producto, encargados de construirlo y operarlo. Sin olvidar trasladar las lecciones aprendidas en estos equipos al conjunto de la compañía. Para ello es indispensable dotar de visibilidad a los equipos y sus integrantes. Aspectos como la formación interna, artículos en el blog, creación de mesas redondas, colaboración y cultura de la comunicación son indispensables para no romper con la libertad tecnológica de cada proyecto, a la vez que se garantiza una línea común de trabajo.

En la compañía se han identificado algunas lecciones aprendidas en el viaje al mundo DevOps, algunas de ellas son:

- **La disposición de recursos y apoyo de todas las áreas de la organización es indispensable.** Tanto dirección, equipos de producto, QA y seguridad, como líderes en metodologías *agile,* que han facilitado la adopción de prácticas DevOps en los equipos.
- **Camino de menos fricción y mayor impacto.** Mediante el uso de técnicas como *Value Stream Mapping* o *Theory of Constraints*, es posible detectar aquellos puntos donde una pequeña mejora (un cambio focalizado y con ámbito limitado) puede aportar un mayor **beneficio**. Además, es importante contar con un equipo dispuesto a mejorar. Estas acciones ayudan a avanzar enormemente en la implantación de DevOps en la compañía.
- Como consecuencia del punto anterior, se concluye que **es preferible implantar prácticas DevOps** tratando de conseguir victorias rápidas, que aporten **beneficio** en el plazo de un mes o dos, en vez de tratar de organizar una gran acción de forma global buscando un gran impacto en toda la compañía.

Además, se ha dedicado esfuerzo a mejorar los **procesos** de construcción de ***software***. Si bien en años anteriores se han automatizado partes concretas del ciclo de vida del ***software***, recientemente se ha comenzado a observar cómo el ciclo completo comenzaba a madurar. Los equipos han automatizado flujos de trabajo, permitiéndoles aumentar la frecuencia de ***release*** del ***software***, la calidad y disminuir coste, defectos y tiempo necesario para solucionar errores. Los equipos han compartido razonablemente un variado ecosistema de herramientas y conceptos. Esta forma de construir ***software*** refleja, según la Ley de ***Conway***, la estructura de comunicación de la compañía. Desde el punto de vista **tecnológico**, se puede decir que el foco se ha situado sobre las siguientes áreas:

- ***Continuous Integration/Continuous Delivery:*** es la práctica que permite aumentar la cadencia de entrega de valor en sistemas productivos a la vez que se aumenta o mantiene la calidad, seguridad y se reduce el coste, los tiempos de recuperación y los errores. Actualmente, el foco está puesto sobre la correcta orquestación de los ***pipelines***. Jenkins 2.0 se ha establecido como el rey en este área, ya que permite el uso de **Jenkinsfile.** De este modo es posible definir ***pipelines*** como código y programar una lógica más adecuada y adaptada a cada caso particular.
- **Infraestructura como código**: sin duda la infraestructura como código es un valor clave para el éxito de cualquier entorno. Esta práctica se ha extendido en los principales proveedores de *cloud* utilizados (AWS **Cloudformation**, Azure Resource Manager Templates y Google Deployment Manager Templates). Así como **Terraform**, que se alza como el rey en escenarios multi-nube. Concretamente, se han realizado numerosas mejoras en lo que a gestión de parámetros (de forma segura), diseño de infraestructuras débilmente acoplada y despliegues Blue/Green se refiere. Dicha evolución ha permitido mejorar la reutilización, la seguridad y reducir el impacto de los despliegues en producción en términos de disponibilidad. Ha sido interesante observar cómo perfiles de desarrollador ***software*** se han involucrado activamente en el desarrollo de la infraestructura como código, dejando de ser un nicho exclusivamente de perfiles de administradores de sistemas.
- **Plataformas *cloud***: el ***cloud computing*** es sin duda una tecnología que habilita y permite que las prácticas DevOps alcancen su mayores cuotas. Aunque **AWS** sigue siendo el líder en cuota de mercado, otros proveedores como **Azure** o **Google Cloud** están avanzando a muy buen ritmo y a día de hoy se consideran competidores de pleno derecho. Además, ambos proporcionan funcionalidad enfocada a gestión del cambio y ***pipelines,*** por lo que muestran su capacidad para participar en la implementación de metodologías DevOps. Además, otras plataformas como **Kubernetes** han demostrado su liderazgo formando parte de las tecnologías que, sin duda, están siendo adoptadas. Tanto es así, que la mayoría de proveedores ***cloud*** lo soportan por completo, incluso aquellos que inicialmente ofertaban su propia solución, como AWS ECS.
- **Gestión de la configuración**: a pesar del trabajo realizado hasta ahora automatizando la configuración de la infraestructura, principalmente con **Ansible**, este año se ha avanzado en un aspecto concreto de la misma. Este aspecto es la gestión adecuada de los valores de configuración y su disponibilidad. Habitualmente, durante un despliegue de infraestructura o aplicación, es necesario crear una configuración. Cuando dicha configuración es estática se puede considerar almacenarla en repositorios de código, pero esta aproximación no funciona en entornos productivos con un mínimo de complejidad. En ocasiones, dicha configuración es única y debe ser establecida de forma dinámica. Otras, en cambio, requiere conocer información sensible que no debe ser almacenada en repositorios de código. Por ello, prácticas como el uso de ***templates***, junto con el almacenamiento de parámetros en servicios preparados para ello como **AWS Parameter Store** o Vault, han demostrado resolver el problema. Estas tecnologías permiten leer los parámetros necesarios y almacenar aquellos generados de forma dinámica y segura durante la ejecución de ***pipelines***, de este modo fases posteriores los pueden utilizar.

Los equipos han automatizado flujos de trabajo permitiéndoles aumentar la frecuencia de release del software, la calidad y disminuir coste, defectos y tiempo necesario para solucionar errores

Próximos pasos

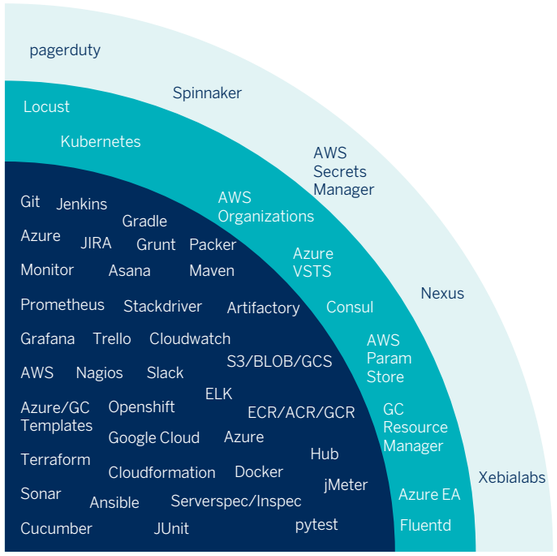
El aspecto más importante cuando se trata de aplicar metodología DevOps es el cultural. Una verdadera transformación requiere tiempo y apoyo de la organización. La **cultura** de una empresa es algo que se debe cuidar a diario y por lo tanto es necesario seguir trabajando en la línea actual. Para ello, es necesario apoyarse en técnicas ***agile*** a la vez que se siguen apoyando aspectos como la integración en los equipos de desarrollo de las tareas relacionadas con las operaciones IT. La convergencia del mundo Agile y DevOps es una realidad, como indican Dave West y Jayne Groll (CEOs de Scrum.org y DevOps Institute, respectivamente) en su artículo *The Convergence of Scrum and DevOps*.

El aspecto más importante cuando se trata de aplicar metodología DevOps es el cultural

Además, también es necesario trabajar en la obtención de ***feedback*** a todos los niveles. Dicho ***feedback*** es necesario para tomar decisiones basadas en datos reales con el objetivo de mejorar. Es algo habitual en el mundo Agile pero, sin duda, tiene su reflejo en el mundo DevOps en forma de encuestas (como el reporte anual del estado de DevOps de Puppetlabs y DORA), medición de métricas como el ***lead time, recovery time***, porcentaje de despliegues fallidos o frecuencia de despliegues.
Iniciativas internas enfocadas a la formación, la creación de mesas redondas de debate, publicación de casos reales de éxito (o fracaso), la colaboración entre personas y la creación de flujos de trabajo seguros donde las personas puedan investigar nuevas opciones y asumir riesgos controlados, son indispensables. Además, bajo el paraguas de la tecnología y gracias a arquitecturas basadas en eventos, es posible asegurar el cumplimiento de normativas y requisitos sin la necesidad de tomar control absoluto de todo aquello que se implementa.
De este modo será posible crear una **cultura colaborativa, libre de miedo a equivocarse** y con un buen grado de **homogeneización tecnológica**. Todo ello sin necesidad de eliminar la libertad tecnológica de los equipos y manteniendo el foco en el desarrollo de funcionalidad para el cliente.
A nivel puramente tecnológico se debe seguir avanzando en infraestructura como código, principalmente en gobierno mediante el uso de servicios como **AWS Organizations, Azure EA o Google Cloud Resource Manager**, que permiten el control de las cuentas ***cloud*** y el control de los recursos desde un punto central. Como puntos a destacar, permiten la creación de políticas y el despliegue de configuraciones de infraestructura predefinidas que cumplan con las políticas de gobierno definidas en la organización.
El elemento de unión de cada fase en el ciclo de desarrollo del ***software*** es el ***pipeline***: es el momento de obtener la madurez completa del mismo. Para ello es necesario hacer hincapié en el uso de **Jenkinsfile** y empezar a considerar el uso de **Shared Libraries**, no solo en grandes proyectos, sino también en aquellos que comienzan.
Implantar y comprender ambos permitirá madurar considerablemente los ***pipelines*** y proveerá a los equipos de la herramienta necesaria para que adquieran autonomía gestionando el ciclo de vida del ***software***. Por último, es necesario comenzar a explorar herramientas de visualización y control de ***pipelines***, como ya ofrecen **Spinnaker** y **Xebialabs**.
Desde el punto de vista concreto de los despliegues, el ecosistema Docker ha madurado enormemente en los utimos años. Por lo tanto, se debería comenzar a adoptar con total confianza. Multitud de opciones han encontrado esta madurez, concretamente AWS ha dado un paso anunciando un servicio que proporciona un ***cluster*** **Kubernetes**, llamado **AWS EKS**. Pero no solamente eso, sino que ha aparecido también **AWS Fargate**, que permite desplegar contenedores sin preocuparse por la infraestructura subyacente. En cambio, no se debería dar de lado aspectos como el ***service discovery*** con **Consul**.

El cloud, los microservicios y herramientas de alto nivel están permitiendo que la interoperabilidad entre herramientas de automatización se dispare, eliminando de este modo el "hazlo tú mismo"

Respecto a la obtención de ***feedback***, el camino parece estar claro. El ***stack*** **ELK** está asentado en el mercado y dentro de la compañía. Además, la incorporación de arquitecturas basadas en eventos con herramientas como **AWS Config Rules**, **AWS Cloudwatch Events** o similares ha de seguirse con mucho interés, pues pueden ser la llave del nuevo gobierno de cualquier nube, favoreciendo un flujo de trabajo que otorgue libertad a los equipos sin descuidar seguridad, auditoría y flujo de trabajo.
Por último (pero no menos importante) servicios como **AWS GuardDuty, AWS Macie** y otros ya existentes como AWS Trusted Advisor proporcionan monitorización continua de aspectos relacionados con seguridad, localización de información sensible y buenas prácticas en AWS. Sin duda es un ejemplo de cómo el concepto **DevSecOps** debe adquirir más protagonismo e implantarse de forma definitiva como una estrategia continua de seguridad, muy distinta a la aproximación tradicional de auditorías periódicas por elementos externos a los equipos.
En definitiva, el ecosistema **DevOps** aún tiene mucho que decir, el cambio cultural está comenzando a acelerarse tras un tiempo de aceptación del cambio en las personas. En esto ha sido clave la creación de una masa crítica de personas que aceptan el cambio fácilmente.
El ***cloud***, los microservicios y herramientas de alto nivel están permitiendo que la interoperabilidad entre herramientas de automatización se dispare, eliminando de este modo el "hazlo tú mismo" que impedía en muchos equipos conseguir un grado de automatización real, y eliminando la excusa para seguir manteniendo elementos tan importantes como la seguridad al margen de la transformación. ■



RADAR2018
DevOps

Core Adoptar Explorar

BBVA
Next Technologies

09. Machine learning

El equipo de científicos de datos ha crecido notablemente debido a un incremento en el número de **proyectos relacionados con *machine learning*** y de proyectos de exploración de datos

Estado actual

Los últimos años han sido intensos para la ciencia de datos en BBVA Next Technologies. El equipo de científicos de datos ha crecido notablemente debido a un incremento en el número de proyectos relacionados con *machine learning* y de proyectos de exploración de datos.
Nuestras "armas básicas" siguen siguen siendo las mismas y, aunque **Python** y sus magníficas herramientas han seguido ganando importancia en detrimento de **R**, seguimos usando intensamente ambos *frameworks* . A lo largo del año pasado incluimos en nuestro "arsenal" varias herramientas muy potentes e interesantes que comentamos a continuación.

_Productivización de modelos

La necesidad de **productivizar** modelos evitando los infiernos de dependencias, los problemas de integración o la reescritura de los modelos en lenguajes compilados, nos ha llevado a adoptar **Docker**, junto con **Plumber** o **Flask** , como tecnología *core* en varios proyectos, con excelentes resultados.
La tendencia a disponibilizar modelos como APIs que comentamos hace un año se ha hecho realidad, y ésta se ha convertido en una forma muy frecuente de productivizar modelos.

La tendencia a disponibilizar modelos como APIs que comentamos hace un año se ha hecho realidad, y ésta se ha convertido en una forma muy frecuente de productivizar modelos

Hemos añadido al radar la recién aparecida **AWS Sagemaker** como herramienta a explorar. Se trata de un nuevo servicio de Amazon, enfocado precisamente a la productivización de modelos y con una estrategia similar a la que hemos ido evolucionando internamente.

_Nuevas herramientas de visualización

También incluimos en nuestro radar herramientas no directamente relacionadas con *machine learning* , pero sí vinculadas con otras áreas de la ciencia de datos. Así, para visualización interactiva de resultados, **Plotly** y **Shiny** (a veces usados conjuntamente) han resultado de gran utilidad, y **Leaflet** se ha impuesto como la mejor solución para la representación de datos geográficos.

_Deep Learning

El *deep learning* también ha tenido una importancia destacada en BBVA Next Technologies. **Keras**, **Pytorch** y **MXNet** han sido las herramientas que más nos han convencido (en ese orden). Tras numerosas pruebas, y a pesar de su inmensa tracción y su enorme comunidad, vemos un inconveniente que hay que tener presente a la hora de utilizar **TensorFlow** en proyectos, ya que su API está en constante cambio y es difícil mantener el código y su compatibilidad con otras integraciones.
No obstante, y precisamente por su tracción, creemos que debemos incluir este *framework* en el área de tecnologías a adoptar, de cara a fomentar su conocimiento dentro de la empresa y poder afrontar proyectos que lo exijan como requisito. Añadimos **Caffe2** como tecnología a explorar para la productivización de modelos, y **ONNX** como estándar de serialización genérico de modelos de *deep learning*. Esta herramienta nos permitiría, por ejemplo, convertir modelos **Pytorch** (más cómodos para el científico de datos pero menos eficiente para poner en producción) en modelos de **Caffe2**.
Los desarrollos en *deep learning* han ido muy de la mano de la evolución de las **GPU**, que han sido notables y que hemos seguido muy de cerca. Hemos realizado numerosas pruebas para entrenar modelos con una o varias GPU usando

instancias de **AWS** y **GCP**. También hemos explorado el uso de las GPU en el contexto de BBDD, con tecnologías como MapD, y ya estamos usando GPUs en varios proyectos basados en *frameworks* de *deep learning*.
También hemos probado herramientas como **Floydhub** y **Google Cloud MLE**. PaaS que abstraen al científico de datos de las dificultades de instalar y configurar un entorno de experimentación con librerías como **Tensorflow,** permitiendo además la ejecución sobre GPU.
Otras soluciones que han surgido en este ámbito tienen que ver con los contenedores. **NVIDIA** , en colaboración con Amazon, ha sacado unos ***deep learning containers*** que usan tecnología derivada de Docker para proporcionar las últimas versiones de los *frameworks* de *deep learning* más populares, completamente optimizadas por los ingenieros de NVIDIA y preparadas para un uso fácil y rápido.
Otra novedad reciente ha sido la aparición de numerosos recursos de gran calidad (gratuitos y de pago) para aprender *deep learning*, explicando esta tecnología desde cero hasta los últimos avances. Entre los cursos, destacamos el de **deeplearning.ai** (con un enfoque más teórico, de Andrew Ng); el de **fast.ai** (con una orientación más práctica y apoyado en el framework PyTorch); y varios "nanodegrees" de Udacity con fuerte contenido relacionado con el *deep learning* (y el *machine learning* en general), como el **Deep Learning Nanodegree Foundation** o el **Self-Driving Car Engineer Nanodegree**, todos muy completos y que son actualizados continuamente.

Otra novedad ha sido la aparición de numerosos recursos de gran calidad (gratuitos y de pago) para aprender deep learning, explicando esta tecnología desde cero hasta los últimos avances

_Otros

Gracias a dos proyectos relacionados con la **detección de anomalías en series temporales**, el equipo ha acumulado una gran cantidad de conocimiento en este campo, lo que lo ha convertido en una nueva línea estratégica para BBVA Next Technologies.
Para la solución de problemas muy concretos que forman parte de proyectos más grandes, se han utilizado con éxito tanto **Microsoft Cognitive** como **Google Cloud AI**. Dentro de estas dos *suites* de APIs de Inteligencia Artificial cabe destacar la tecnología de OCR integrada en Google Cloud Vision, usada primero en pruebas de concepto y posteriormente en proyectos en cliente, así como la herramienta **LUIS** de Microsoft para hacer tareas de NLU, que hemos usado para el desarrollo de distintos *chatbots*.
H2O sigue siendo una herramienta muy interesante pero a la que no hemos dado demasiado uso. La mantenemos como herramienta a adoptar.
En el área de Visión Artificial hemos utilizado para varios casos de uso la librería **Dlib** con muy buenos resultados. Sin embargo, es una librería muy amplia y que cuenta con algoritmos y herramientas no necesariamente relacionadas con *machine learning*. Hemos decidido adoptar la parte de Dlib directamente relacionada con ML pero seguir explorando el resto, porque creemos que es una herramienta con potencial para usar en nuestro día a día.
En este contexto, hemos probado **Amazon Rekognition** en tareas relacionadas con reconocimiento facial. Es una herramienta muy cómoda y sencilla de utilizar, pero que carece de la flexibilidad que aporta, por ejemplo, Dlib. Recomendamos su uso si los datos ya se encuentran en AWS o si se carece de perfiles especialistas en *machine learning*.

Creemos que una formación básica en arquitectura dará una perspectiva más amplia a los científicos de datos a la hora de orientar todo el pipeline a la productivización final de los modelos

Gracias a nuestro trabajo con **GPUs** de **NVIDIA**, llegamos a un acuerdo con la asociación AILoveU y con el principal proveedor de NVIDIA en España, la cual nos permitió explorar la tecnología de supercomputación **NVIDIA DGX**. Esta solución *hardware* y *software* podría tener su hueco dentro de nuestras propuestas tecnológicas para su uso en proyectos de *deep learning* que, por la confidencialidad de los datos, hagan imposible el uso de las soluciones *cloud*. Hablamos de proyectos que requieran de una potencia de cálculo masiva a la que no podemos responder con otro tipo de *hardware commodity*.
Rasa NLU es una herramienta que hemos estudiado a fondo. Siendo muy interesante, ha obtenido por nuestra parte el calificativo de "objetivo en movimiento" con lo que recomendamos su

uso con precaución. Si bien es fiable y resuelve satisfactoriamente los problemas que le hemos planteado, el hecho de que esté teniendo tantos cambios y tan rápidos, hace que solo la aconsejemos para proyectos en los que, nuevamente, la confidencialidad de los datos imposibiliten el uso de herramientas *cloud*.
Algunos compañeros del área de ciencia de datos han conseguido certificaciones como **Google Cloud Data Engineer** o **AWS Certified Solutions Architect - Associate**. Creemos que una formación básica en arquitectura dará una perspectiva más amplia a los científicos de datos a la hora de orientar todo el *pipeline* a la productivización final de los modelos.

Próximos pasos

Actualmente seguimos explorando nuevas tecnologías y áreas de aplicación. Dedicaremos este espacio a hablar de tecnologías potenciales en el contexto de diferentes áreas estratégicas.

_'Fairness', interpretabilidad y regulación

Con el avance de la regulación europea en temas de privacidad, hay cada vez un interés más generalizado en explorar formas de hacer los modelos más transparentes y más justos. En este contexto, seguiremos atentos a los avances científicos en materias de "interpretabilidad" y *fairness* para poder tener una ventaja competitiva al aplicar modelos más complejos en casos sensibles.
En cuanto a interpretabilidad están surgiendo numerosas propuestas (LIME, **SHAP** y otros) para interpretar, al menos parcialmente, modelos muy complejos (especialmente en el campo del *deep learning*). También resurgen antiguas herramientas estadísticas para interpretar modelos (PDPs, ICEs,...) que se están adaptando con mucho acierto para su uso en *machine learning*.
En cuanto al *fairness* , el problema ya pasa a ser no interpretar una predicción, sino controlar los factores inherentes a los datos que se usan para hacer esas predicciones y, finalmente, poder medir y controlar las fuentes de una posible discriminación injusta. Estamos viendo muchos avances e interés también en este área y, combinados con el área de interpretabilidad, pueden ayudarnos a adaptarnos más a las nuevas regulaciones que vayan surgiendo.

También resurgen antiguas herramientas estadísticas para interpretar modelos (PDPs, ICEs,...) que se están adaptando con mucho acierto para su uso en machine learning

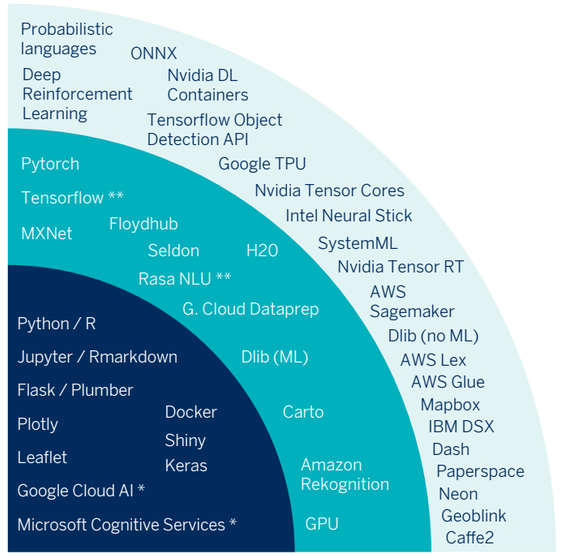
En BBVA Next Technologies seguimos con interés estos avances y estamos ya preparando formación específica para que sean aplicados e integrados en todos los proyectos que incluyan procesos de *machine learning*. Esto permitirá que nuestros científicos de datos generen modelos de mayor calidad, e incrementará la confianza de nuestros clientes en los modelos que creamos.

_Deep Learning

Los rápidos avances que están ocurriendo en las fronteras del *deep learning* son de mucho interés para BBVA Next Technologies. La comunidad *opensource*, apoyada por grandes empresas como Google, Facebook, Intel o NVIDIA, cada vez propone o apoya *frameworks* más usables o más eficientes. En este contexto, *frameworks* incipientes como **Caffe2** o Neon, que pretenden avanzar hacia la eficiencia y escalabilidad, o **Pytorch**, que avanza hacia la usabilidad, merecen nuestro interés de cara a productivizar o desarrollar modelos de *deep learning*.
Si bien veíamos que NVIDIA había trabajado en tecnologías para el entrenamiento, no ha dejado de lado la productivización y cuentan con **Tensor RT**, una herramienta para comprimir modelos y optimizarlos, haciéndolos más rápidos y ligeros. Esta herramienta, en combinación con su nueva solución *hardware*, **Tensor Cores** (dentro de las nuevas GPU Volta), permite acelerar la inferencia órdenes de magnitud.
Aunque, por contra, ata esos modelos a su uso exclusivo en las GPU de NVIDIA, lo que dependiendo del caso de uso no es viable, como por ejemplo, al trabajar con tecnologías *serverless*.
Por otro lado, Google también ha estado trabajando para mejorar la productividad y la inferencia con su *hardware* especializado para redes neuronales llamado **TPU** (Tensor Processing Unit), que fue anunciado como parte de su estrategia *cloud*.
Por otra parte, está habiendo un gran interés de la comunidad científica por modelos probabilísticos en el contexto del *deep learning*. Esto es, modelar cosas como la distribución de probabilidad en vez de dar directamente una predicción, permitiendo aplicaciones como la generación de datos "realistas" o el modelado de la incertidumbre en cada predicción. Hasta ahora esos son problemas abiertos, y en ese contexto surgen (o resurgen) algoritmos como *generative adversarial* networks (**GANs**), o redes bayesianas.
También surgen *frameworks* denominados ***Probabilistic Programing Languages,*** que permiten resolver algunos de estos problemas de una manera relativamente sencilla, cuya aplicabilidad y usabilidad queremos estudiar. Un buen ejemplo es Pyro, el *framework* desarrollado por Uber para este propósito.

_Reinforcement Learning

A medida que las técnicas de inteligencia artificial avancen y sean más adoptadas por el sector tecnológico, no es de extrañar que cada vez se afronten problemas más complejos. Uno de ellos es el del aprendizaje automático de la toma de decisiones en sistemas dinámicos complejos, comúnmente referido como el problema de ***Reinforcement Learning***.
En la comunidad científica estamos viendo cada vez avances más impresionantes en este área, especialmente en su intersección con las técnicas de *deep learning*. Varias empresas españolas y de todo el mundo ya están explorando las potenciales aplicaciones de este área. Asimismo, en BBVA Next Technologies hemos participado en la exploración de estas tecnologías de cara a varias aplicaciones, y es por ello que las consideramos un área a seguir explorando. ■



RADAR**2018**
Machine Learning

* : Se han usado y testeado algunos servicios
** : Adopción con precaución. "Objetivo en m

Core Adoptar Explorar

BBVA
Next Technologies

10. Human Computer Interaction (HCI)

Las **apps** cada vez muestran más síntomas de agotamiento en el mercado, y **desarrolladores y creativos quieren descubrir los nuevos canales de comunicación** que mantendrán vivo nuestro interés por el **uso de sus servicios**

Casi cada año aparecen nuevas ideas, conceptos o paradigmas de interfaz humano-máquina. Las *apps* cada vez muestran más síntomas de agotamiento en el mercado, y desarrolladores y creativos quieren descubrir los nuevos canales de comunicación que mantendrán vivo nuestro interés por el uso de sus servicios.
En años anteriores se ha producido el auge de los *chatbots*, que ofrecían como ventaja eliminar la barrera que, cada vez más, supone el proceso de instalación de una nueva *app*. El planteamiento es sencillo: utilizar las *apps* de mensajería que el usuario ya tiene instaladas para ofrecer nuestros servicios.

Estado actual

_Realidad virtual

Durante los últimos años, la realidad virtual y sus dispositivos relacionados han alcanzado, gracias sobre todo a dispositivos móviles y sus sensores de movimiento, un estado de madurez que permite usarlos para diferentes casos de uso. Es por esto que en los Labs de BBVA Next Technologies hemos dedicado esfuerzos en la investigación y desarrollo de diferentes tecnologías y las herramientas que permiten trabajar con ellas.
De todas ellas, nos hemos centrado en el desarrollo para las 3 plataformas con más penetración en el mercado, como son **HTC Vive, Oculus Rift** y Google **Cardboard/ DayDream**.
En la actualidad, para desarrollar en cualquiera de las dos primeras plataformas, se puede optar por el motor de Unity, basado en c#, el motor de Unreal basado en C++ o el motor de Lumberyard, desarrollado por Amazon como **IDE** y motor multilenguaje.
Lumberyard, a día de hoy, carece de muchos de los SDK para diferentes dispositivos, por lo que actualmente nos hemos centrado en desarrollos con **Unity** y **Unreal**.
Ambas ofrecen excelentes resultados y disponen de SDK para el desarrollo en todos los dispositivos de VR como HTC Vive, Oculus Rift, DayDream, dispositivos de ***mixed reality*** bajo el estándar de Windows, etc. No obstante, hemos optado en más ocasiones por el desarrollo bajo Unity, porque ofrece un mayor control sobre el rendimiento de la aplicación en dispositivo y mayor versatilidad a la hora de crear componentes y módulos propios.

_Realidad aumentada

En el área de la realidad aumentada, el año 2017 ha sido el año de **ARKit** y **ARCore**. Las dos soluciones de realidad aumentada para dispositivos móviles de Apple y de Google respectivamente, muy similares en cuanto a concepto y tecnología, han supuesto la democratización del uso de la realidad aumentada, como lo fueron hace 4 años las Cardboard de Google.
Es por eso que hemos centrado esfuerzos en desarrollar para ellas y analizar pros y contras. Ambas, como mencionamos, han contribuido a la democratización del uso de la AR, al prescindir de marcadores físicos para la ubicación de modelos virtuales en el entorno real. En su lugar, se sirven de una muy básica librería de visión artificial que localiza planos horizontales donde poder fijar los modelos virtuales.
En este punto las tecnologías de inteligencia artificial confluyen con la realidad aumentada. Y es que, tecnologías como la visión artificial basada en aprendizaje por refuerzo son el pilar sobre el que se sustenta la futura integración de elementos virtuales en el entorno real y los objetos que lo componen.

_Chatbots y conversacionales

En cuanto a los motores de entendimiento del lenguaje (NLU), aquello que hace que los ***chatbots*** nos entiendan, hemos visto la consolidación del modelo de intenciones, entidades y frases de entrenamiento. Se han realizado experimentos, sobre todo con las dos plataformas mayoritarias, **Wit.ai** y **Dialogflow** (ex Api.ai), probando que las dos están listas para producción, por lo cual, su

estado pasa a la fase de "Adopción". La tercera herramienta del radar, **Recast.ai**, se mantiene por explorar. Inicialmente se había descartado por su bajo soporte para el castellano y su nueva estrategia orientada a convertirse en un gestor de ***chatbots***. Sin embargo, la reciente evolución de su servicio de NLU, sobre todo el soporte mejorado al idioma castellano y la división de la plataforma en distintos productos, hace interesante el mantenerlo en el estado de "Explorar" para volver a evaluarla.

Hemos podido validar también distintas bibliotecas y frameworks con la que hemos realizado varios chatbots disponibles para el público

Hemos podido validar también distintas bibliotecas y ***frameworks*** con los que hemos realizado varios ***chatbots*** disponibles para el público. Esta experiencia nos ha permitido descartar la tecnología **Hubot** y validar el uso de **Botkit**.

Por otro lado, en cuanto a los canales, hemos profundizado en el uso de **Facebook Messenger, Slack** y **Telegram** como plataformas para ***chatbots***. Las características y ***features*** de interacción de estos canales permiten desarrollar una interacción muy completa. Tenemos desde pasarelas de pago hasta funcionalidades de NLP directamente integradas en las plataformas de comunicación. Por otro lado, las APIs son maduras, estables y bien documentadas. También mencionar que la utilidad de cada una de ellas es relativa al público al que se quiere dirigir la aplicación. Orientando Slack a ***chatbots*** de productividad y entornos profesionales, y Telegram y Facebook Messenger a ***consumers*** generalistas, hemos tenido en cuenta que cada uno de ellos utiliza sus propios medios, como carruseles o teclados personalizados. Sin embargo, para utilizar toda la funcionalidad de estos canales de forma sencilla es necesario utilizar bibliotecas como **Claudia Bot Builder** o **MS Bot Framework**, que deberemos explorar en el futuro.

En cuanto a asistentes físicos, desde los Labs de BBVA Next Technologies hemos realizado varias pruebas con **Amazon Alexa** y **Google Home**. Se ha visto que aunque la barrera de entrada es más pequeña en Alexa, usando su entorno de desarrollo está especialmente orientada a transacciones, como pedidos a domicilio. Por otro lado, Google Home se ha visto que está más orientado a mantener una conversación, aunque la barrera de entrada sea mayor, ya que hay que implementar toda la lógica conversacional. En cualquiera de los casos, parece orientado al mercado doméstico, adoleciendo de falta de sistemas de gestión para grandes volúmenes de dispositivos.

Próximos pasos

_Realidad virtual

Desde febrero de 2018, disponemos del nuevo estándar de desarrollo de proyectos de realidad virtual web. Con **WebVR** no solo disponemos de un estándar al que se han adherido todos los principales navegadores, sino que dispone de librerías de acceso a diferentes dispositivos de interacción como los ***sticks*** de HTC Vive, de Oculus Rift o controladores Bluetooth directamente desde el propio navegador.

Incluso gigantes como Amazon han apostado por el futuro de la realidad virtual en plataformas web, y están trabajando en un servicio de desarrollo de VR *cloud*: **AWS Sumerian**. En este servicio hay un interfaz gráfico para desarrollar y exportar un proyecto bajo el estándar WebVR.

Bajo la misma premisa de la realidad aumentada de priorizar la democratización del uso de la tecnología, la tendencia durante los próximos años parecen ser las aplicaciones VR Web, que permiten desarrollos multiplataforma accesibles para una mayor cuota de usuarios con un rendimiento en las aplicaciones que mejora con cada actualización de los principales navegadores.

Desde febrero de 2018, disponemos del nuevo estándar de desarrollo de proyectos de realidad virtual Web

En BBVA Next Technologies ya hemos trabajado en muchas ocasiones con **Three.js**, librería de WebGL que renderiza en navegador vectores tridimensionales, pero esta librería ha experimentado una gran evolución adaptándose al estándar y es uno de los principales motores sobre el que trabajan otras tecnologías como A-Frame o AWS Sumerian. Por lo que consideramos una gran oportunidad ampliar nuestro conocimiento sobre ella, así como sobre AWS Sumerian como sistema de desarrollo mediante interfaz.

_Realidad aumentada

Al igual que en realidad virtual, en realidad aumentada también han aparecido soluciones basadas en navegadores web, aunque en este caso, su uso debe responder a un estudio previo del caso de uso y necesidades, ya que dependiendo de si una aplicación AR creada para web se abre en un móvil o en un ordenador de sobremesa, el comportamiento será completamente diferente.

La realidad aumentada se sirve de los sensores de los dispositivos móviles a la hora de realizar el tracking

La realidad aumentada se sirve de los sensores de los dispositivos móviles a la hora de realizar el ***tracking***, si se despliega en un ordenador de sobremesa perdería esa funcionalidad. Aun así tecnologías como AR.Js están evolucionando rápido y merecen un análisis para su aplicación en casos de uso muy específicos.
Por otro lado, tanto **ArKit** como **ARCore** están evolucionando a una gran velocidad, añadiendo funcionalidades cada poco tiempo y mejoras en el área del posicionamiento absoluto y análisis del entorno, que amplían sus casos de uso, por lo que su análisis es una constante durante este año, como lo fue el año pasado.

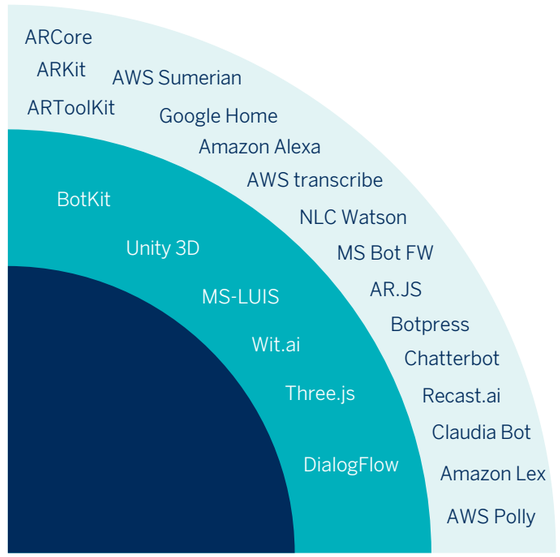
_Conversacionales

Está previsto que salgan al mercado las versiones **en español** de varios de de los asistentes de voz, como **Alexa** y **Google Home**. Será necesario evaluar cómo cubren este idioma.
También es necesario evaluar herramientas que permitan una gestión empresarial de los dispositivos asociados a estos asistentes, por lo que soluciones como **Alexa for Business** son necesarias para poder afrontar proyectos de cierta envergadura.
En cuanto a ***frameworks*** para gestión de diálogos, herramientas como **Dialogflow** han añadido memoria y contextos en las conversaciones. Es necesario continuar explorando qué **nuevas funcionalidades** añaden. No obstante, se hace demasiado difícil gestionar conversaciones complejas con estas herramientas, por lo que es necesario hacer un seguimiento sobre nuevas formas de modelar las conversaciones que implementen estas herramientas.
Los **canales** han ido añadiendo **nuevas formas de interacción** con los usuarios que permiten enriquecer y facilitar la relación con estos. En este

Sobre la asimilación de las interfaces conversacionales en el mercado, la mayoría de las grandes marcas ya han estado buceando en estas nueva forma de interacción y parece que en su mayoría han chocado con sus limitaciones

sentido, es necesario continuar explorando los nuevos elementos que se introducen y su acogida por parte de los usuarios.
La comunicación entre humanos es altamente dependiente del **contexto** en que se da: tanto a nivel de la propia conversación, como de los elementos que la rodean, como ubicación, hora o experiencias previas. **La inclusión de los contextos dentro de la experiencia conversacional mejorará mucho la comprensión** de las intenciones de los usuarios y es necesario explorar su potencial.

Sobre la asimilación de las interfaces conversacionales en el mercado, la mayoría de las grandes marcas ya han estado buceando en estas nueva forma de interacción y parece que en su mayoría han chocado con sus limitaciones. Cada vez parece más clara la necesidad de incorporar varios dispositivos en la interacción, aprovechando las ventajas de cada uno. De esta manera, la llamada **omnicanalidad**, o multimodalidad, parece que será una línea de investigación muy importante. ■



BBVA
Next Technologies

11. Seguridad 2018

Las **tecnologías que van a tener impulso** durante los próximos años **son aquellas que mejoran la experiencia de usuario, la gestión o** proponen una **evolución** en la forma de detección y nuevos mecanismos de protección

Estado Actual 'Workplace Security'

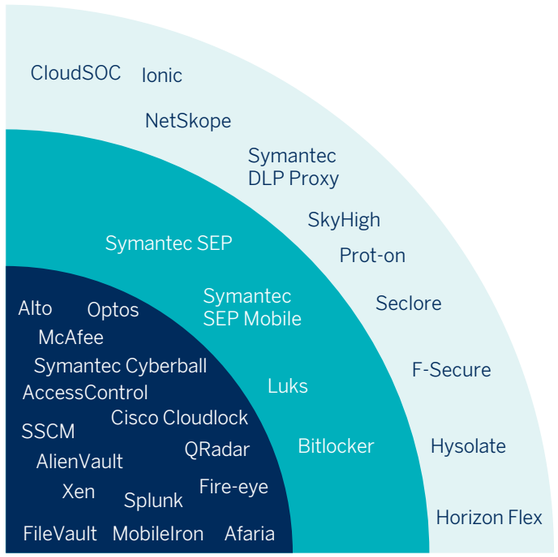
Para BBVA Next Technologies la seguridad en el contexto del entorno de usuario se compone principalmente de cuatro conjuntos tecnológicos: ***Data Loss Protection*** **(DLP),** ***antimalware*****, control de acceso y bastionado de equipos.**
La parte **DLP** está formada por aproximadamente 19 tecnologías con el único objetivo de evitar o mitigar los 17 vectores de fuga de información que son materializables.
Para ello se encuentran infraestructuras de escritorio virtual (VDI) **Citrix XenDesktop** y virtualización de aplicaciones que se ejecutan en un entorno seguro con **Citrix XenApp**, donde el usuario trabaja en un entorno seguro y limitado para casos donde la información que se maneja es muy sensible. Los ***endpoints,*** además, tienen distintos métodos para proteger los datos, como el cifrado de estos, donde se utiliza una combinación de **Bitlocker, FileVault, Luks y McAfee Encryption** dependiendo del caso de uso. Para evitar la fuga de información a través de dispositivos externos existen controles para el uso de dispositivos USB con tecnologías como **GPO** y **McAfee DLP Endpoint**, para el correo se utiliza **Symantec Cyberball** y sobre recursos compartidos, **Forcepoint AP-Data**.
Para la parte de ***antimalware*** está el grupo Endpoint Detection and Response (EDR) para intrusiones a nivel de *host,* como **HIDS McAfee** y ***antimalware*** de **McAfee**.
En cuanto al **control de acceso** a recursos de la nube, se utiliza **Cisco CloudLock**, que es una tecnología de ***Cloud Access Security Broker*** **(CASB)** que ejerce de gestor de accesos y permisos a distinta documentación que no debería ser accesible según la categoría de esta.
Como parte adicional de la **prevención**, en la parte de bastionado se utilizan tecnologías de bastionado tanto móvil como PC. Dentro de ***Mobile Device Management*** **(MDM) y** ***Enterprise Mobility Management*** **(EMM)** nos encontramos tecnologías de bastionado de dispositivos como **Afaria** y **Mobileiron** para la parte móvil y **SCCM** para bastionado Windows, así como **JAMF** para bastionado MAC.
También existen otras tecnologías que se utilizan pero no impactan directamente sobre el usuario y sus procesos. En cuanto a detección y gestión de alertas dentro de los distintos *workplace* se utilizan distintos ***Security Information and Event Management*** **(SIEM)** para recolectar los eventos que suceden y pueden dar información de un posible ataque y de su escala, como **Qradar, Splunk, AlientVault** e incluso el servicio de **SEMaaS**, que es de la compañía. Mencionar también las herramientas de vigilancia digital que pueden ayudar a "correlar" eventos que ocurren fuera de la compañía, permitiendo anticiparse a posibles vulnerabilidades o ataques: **Alto** y **Optos** son las dos tecnologías que ayudan a ello. Por último, como tecnología **anti-APT** está **Fire-eye**, que detecta posibles *Advanced Persistent Threats* (APT), que son ataques que pueden desestabilizar compañías a través del espionaje, ya que establecen procesos difícilmente detectables.

Próximos pasos

Las tecnologías que van a tener impulso durante los próximos años son aquellas que mejoran la experiencia de usuario, la gestión, o proponen una evolución en la forma de detección y nuevos mecanismos de protección a las ya implantadas.
Para toda la parte de virtualización local se probará **VMware Horizon Flex e Hysolate**, para ver si las ventajas que ofrece a nivel de usuario merecen la pena a la hora de facilitar la gestión de la arquitectura de los escritorios remotos.
Como nueva forma de protección, **Symantec Fireglass** ofrece una alternativa para *web isolation* emulando la sesión web del usuario de forma aislada, enviándole solo la información que tiene un riesgo potencial bajo. Además, añadido al **DLP de *proxy* de Symantec**, se puede tener una ***suite*** de un solo ***vendor*** que controle que toda la navegación del usuario es segura para él y para la empresa.

Como parte de la protección de datos muy sensibles o de riesgo potencial alto, nacen las tecnologías categorizadas como *Information Rights Management* (IRM). En este campo se tiene previsto probar varias herramientas como **Seclore, Prot-on** e **Ionic** para poder adecuar la protección del dato según su categoría.
En el mapa de ***Privilege Access Management*** (PAM) aparece **BeyondTrust**, una empresa con una ***suite*** enfocada a ***password***, servidores y ***endpoints*** donde su punto fuerte es la gestión en un punto único.

Para la parte de ***broker* de acceso a la nube** (CASB) se tiene en el punto de mira tecnologías como **Skyhigh, CloudSOC y Netskope**. La idea es realizar pruebas de concepto para validar que cubren las necesidades que se presentan.
En cuanto a la protección de ***endpoint*** se tiene en mente probar las soluciones de **Symantec** y compararlas con McAfee.
Por parte de las políticas de bastionado se utilizaba SCCM y se quiere estudiar la posible integración con **Microsoft Intune**. ■



RADAR**2018**
Security - Workplace

Core Adoptar Explorar

Gran parte de la **seguridad plataforma** que se desarrolla en **BBVA Next Technologies** está compuesta por productos desarrollados en la compañía

Estado actual 'Platform Security'

Gran parte de la seguridad plataforma que se desarrolla en BBVA Next Technologies está compuesta por productos desarrollados en la compañía, debido a que no existe en el mercado ninguna tecnología que cumpla los requisitos necesarios, ya sea porque está poco madura o no cubre las necesidades de la manera que se requiere.
Los productos desarrollados en la compañía son **Armadillo, LUX y Nauthilus** para autenticación y autorización; Chameleon en la parte de criptografía; Chimera para DevSecOps y Vats para verificación de vulnerabilidades en arquitecturas técnicas.
A continuación vamos a ver con detalle cada uno de ellos. Los grupos tecnológicos pertenecientes a Platform Security son las piezas de seguridad asociadas a la autenticación y autorización, criptografía, almacenamiento de secretos, ***hacking*** ético, ***incident and response*** en ***cloud***, DevSecOps y verificación técnica de seguridad.
En muchos casos no nos basamos en productos existentes en el mercado, sino en componentes ***open source*** que nos proporcionan un punto intermedio entre seguridad, reutilización y flexibilidad. Además de implementaciones concretas (productos) de funcionalidades de seguridad, es vital la observación de los estándares de seguridad asociados.

- **FreeIPA**. Es una solución integrada de identidad que provee autenticación centralizada y autorización, almacenando información sobre usuarios, grupos y máquinas. Se encuentra desplegado en R3 (laboratorio, técnico y producción) y en R4. En el caso de Grasshopper, se está utilizando para almacenar las claves SSH y los ***passwords*** de los usuarios, así como para generar los ***tokens*** Kerberos que hacen que un usuario autenticado en Grasshopper pueda saltar a un servidor final.
- **Forgerock - OpenIDM**. Es un gestor de identidades escrito en Java diseñado para ser flexible y que dispone de diferentes conectores para integrarse con otros servicios. Su código fuente está disponible bajo la licencia *Common Development and Distribution License* (CDDL). Utiliza JavaScript como lenguaje para definir reglas de negocio para el aprovisionamiento. Todas las capacidades de OpenIDM exponen interfaces RESTful, que se han desplegado y probado de forma básica.
- **Shibboleth.** Es la implementación de código abierto de referencia del protocolo de intercambio de información de autenticación y autorización SAML. Forma parte del núcleo del IDP Nauthilus.
- **Authentication Service**. Recibe elementos de información sobre un usuario y responde si la validación de esos elementos de información ha sido satisfactoria o no. En función del servicio de autenticación invocado, realizará la validación mediante un mecanismo de autenticación u otro.

Está implementado en el componente SerVal, íntimamente relacionado con el IdP Nauthilus.

- **Balana (by WSO2)**: Es una implementación ***open source*** de referencia del modelo de autorización externalizada, basada en el estándar XACML (*eXtensible Access Control Markup Language*). Forma parte de WSO IS (*Identity Server*), un producto ***open source*** de gestión de identidades y accesos. También forma parte del autorizador LUX, que está asociado tanto al IdP (Nauthilus) como al proxy de seguridad (Armadillo). LUX es el encargado de la autorización de los sistemas de autenticación para la identificación de usuarios en Nauthilus y, al mismo tiempo, de la autorización del acceso a los servicios protegidos por Armadillo. Esto permite gestionar una única base de políticas de autorización.
- **Spring Security**: *framework* de Spring orientado a la autenticación y autorización de aplicaciones. Las claves de su uso, tanto en Nauthilus como en Armadillo, son la facilidad de extensión (propia de todo el ecosistema Spring) y la abstracción de la complejidad asociada a los protocolos de seguridad involucrados.
- **Estándares de seguridad**: los principales estándares de seguridad asociados a este contexto son los siguientes:
  - SAML (*Security Assertion Markup Language*): estándar de autenticación enfocado en el SSO (*Single Sign On*).
  - OAuth 2: estándar de delegación de acceso orientado a que un sitio web pueda acceder a información en tu nombre sin necesidad de compartir el usuario y contraseña del usuario.
  - OpenID Connect: estándar hacia el que están evolucionando las soluciones de federación de identidades y delegación de la autenticación.
  - XACML (*eXtensible Access Control Markup Language*): estándar que permite la definición de políticas de autorización de acceso en un lenguaje declarativo independiente del negocio.
  - JWT (*JSON Web Token*): estándar de creación de ***tokens*** de acceso que contienen aserciones de seguridad sobre información de usuario.
  - SCIM (*System for* ***Cross-domain Identity Management***): estándar de intercambio automatizado de información entre dominios de identidad y sistemas TI.
  - MDQ (*MetaData Query protocol*): protocolo de intercambio de metainformación utilizado para la gestión de los identificadores de proveedores de servicio e identidad.

En el ámbito de la criptografía priman mucho los conocimientos técnicos sobre el negocio y el dato para saber qué tipo de algoritmo es necesario en cada momento y poder aplicar el más correcto

En el ámbito de la **criptografía** priman mucho los conocimientos técnicos sobre el negocio y el dato, para saber qué tipo de algoritmo es necesario en cada momento y poder aplicar el más correcto. Para estos objetivos está **Chameleon**, desarrollado en Golang y utilizando librerías de criptografía de referencia.

- ***As a Service***: con las nuevas tendencias de ***cloud*** surgen nuevos paradigmas sobre la criptografía y la capacidad de ofrecerla como servicio. Actualmente se da servicio:
  - Operaciones criptográficas como HMAC, cifrados simétricos y asimétricos, enmascaramientos, cifrados manteniendo el formato, cálculos de ***hashes*** y firmas o verificación de las mismas mediante distintos algoritmos que ofrecen librerías como **BouncyCasttle** y **Sun JCE**.
  - Para la gestión del ciclo de vida de los certificados se utiliza la tecnologías de **Openssl**.
  - Gestión de claves (almacenamiento y rotado) y almacén de secretos de forma segura (integración con tecnologías de almacenamiento de secretos como **Vault**).

Además se debe prestar especial atención a los distintos estándares criptográficos, ya sea porque pueden dejar de ser seguros, tener debilidades o colisiones, y hay que saber para qué se usan los distintos estándares:

- La gestión de claves donde se utilizan los estándares de representación PKCS5 (ASN.1, DER) y PKCS12(PEM). También se pueden representar mediante el estándar NIST JOSE (JWK).
- Los cifrados simétricos más utilizados en estos momentos son AES, con distintos tamaños de clave y modos (CBC, ECB), y TDES con distintos tamaños de clave (triple y doble clave) y modos (GCM, ECB, CBC). Además, a ambos se les puede añadir el ***padding*** usado actualmente en el estándar PKCS5.
- En los cifrados asimétricos con claves RSA (1024,2048 y 4096) (PKCS1), se utiliza el estándar de cifrado PKCS1 (v1.5 y OAEP).
- Para firma y verificación se sigue el estándar PKCS1 y PKCS7 (certificados digitales).
- En PKI se siguen los estándares PKCS7 (gestión de claves RSA en certificados digitales) y PKCS9 (CSR).
- Además, hay otro tipo de operaciones criptográficas como FPE, que mediante un algoritmo de restricción de caracteres utiliza todos los cifrados simétricos disponibles en el sistema para realizar un cifrado manteniendo el formato.
- En ***Hash***, se utilizan algoritmos diseñados por el NIST SHA1 (obsoleto) y SHA2 : SHA-224, SHA-256, SHA-384, SHA-512.
- Para MAC se usa HMAC, UMAC y CMAC (todas ellas con apoyo de las funciones HASH derivadas de SHA2).

En cuanto a **almacenamiento de secretos** se utilizan tecnologías del tipo HSM (*Hardware Security Module*) o SSM (*Software Security Module*). La utilización de un tipo u otro depende de la legislación de cada país.

Freja es una tecnología de tipo HSM disponible como ***appliance*** independiente y ahora como solución contenerizada. Actualmente se encuentra en estado productivo en R2 como ***appliance*** físico, pero su evolución pasa por el cambio a la solución virtualizada. Permite la generación y almacenaje de ***tokens*** de ***software*** utilizados para autenticación en dispositivos móviles. Permite tanto validación

En la parte de forense e incident response en cloud se está trabajando en la línea de integrar distintas tecnologías bajo el paraguas del conocimiento que se tiene sobre las necesidades a cubrir

de ***tokens*** OTP (*One Time Password*, o "contraseña de un solo uso") como OCRA (OATH *Challenge Response Authentication*, autenticación por desafío respuesta).
Vault es un ***software*** desarrollado por Hashicorp como HSM virtual. Es utilizado por la infraestructura de Meigas y por Chameleon. No es un almacén de secretos al uso, sino un ***middleware*** que cifra y lo almacena en una base de datos de clave valor, como es Consul, también de ***Hashicorp***. Tiene mecanismos de autorización para segmentar los datos por cliente y tiene procesos de sellado y desellado como un HSM tradicional. Además, puede ser utilizado como PKI y como generador de ***tokens*** pseudoaleatorios y controlar el ciclo de vida de los mismos.
En el ***hacking* ético** no se depende exclusivamente de la tecnología, sino de las habilidades humanas para realizarla, al tratarse de un trabajo eminentemente creativo. Las tecnologías ayudan a automatizar determinadas tareas de *hacking* a bajo nivel, pero la diferencia difiere en la habilidad manual.

- **Tecnologías y distribución "bajo uso"**: analizador de protocolos **Web Burp Suite, ZAP Proxy, Dirb**, distribución de *hacking* **Kali**, escaneadores (**Acunetix, Nessus y OpenVas**), **Wireshark** y **Maltego Pro**.
- Actualmente se trabaja en la automatización de pruebas de ***pentesting*** mediante desarrollo propio de herramientas. Por ejemplo, el desarrollo de una herramienta de ***fuzzing*** para la evaluación de la seguridad de APIs definidas mediante formato RAML.

En la parte de **forense e *incident response*** en ***cloud*** se está trabajando en la línea de integrar distintas tecnologías bajo el paraguas del conocimiento que se tiene sobre las necesidades a cubrir.

- **Tecnologías "bajo uso"**: se utilizan todas las funcionalidades de monitorización y rastreo definidas en cada uno de los proveedores *cloud* "bajo interés" (AWS, Google Cloud y Azure).
- Actualmente se trabaja en la automatización de procesos de Incident Response y forense en entornos ***cloud*** mediante desarrollos propios, extendiendo la funcionalidad ofrecida por las API de cada proveedor ***cloud*** y simplificando el uso de herramientas forenses clásicas existentes en distribuciones como Kali Linux, Caine o DEFT.

En cuanto a DevSecOps, **Chimera** es el producto que orquesta la seguridad en tiempo de desarrollo y despliegue, tanto para código fuente como para imágenes *docker*, desarrollado en Python y con algunos servicios en otros lenguajes como Nodejs. Para la revisión de código se tienen las siguientes tecnologías:

- **CheckMarx y Fortify**: los dos gigantes del mercado de análisis estático de código, tienen una gran gama de lenguajes que revisan, aunque las formas en que realizan los análisis y el set de reglas son distintos.
- **Bandit**: tecnología ***open source*** de análisis estático del código de Python2 y Python3.
- **Sonarqube**: se podría llegar a utilizar la herramienta para hacer análisis de seguridad pero se requiere de un *plugin* que detecta OWASP Top 10, SANS Top 25 y CWE, aunque tiene una gama reducida de lenguajes.

Y para el análisis de **seguridad en imágenes y contenedores *docker*** se seleccionaron:

- **Twistlock**: es la solución corporativa para análisis de vulnerabilidades de imágenes y contenedores ***docker***. Se usa mediante el producto de la compañía Chimera, tanto desde ***pipelines*** como usando el frontal de Chimera.
- **Claire**: es una solución ***open source*** de CoreOS orientada a hacer análisis de vulnerabilidades de imágenes Docker y Appc.

Además, estamos utilizando la característica/forma de trabajo de desarrollo guiado por comportamiento (***BDD - Behavior-Driven Development***), para definir las historias de usuario correspondientes a tipos de comportamiento que deben cumplir nuestras aplicaciones desde el punto de vista de seguridad. En este sentido, hemos desarrollado mediante el uso de este método un ***framework*** que contiene varias herramientas de seguridad que pueden ser parametrizadas para ser ejecutadas contra nuestro aplicativo. Nuestro BDD Security emplea para realizar las pruebas de seguridad, herramientas ampliamente reconocidas en la comunidad ***open source*** como: OWASP Zed Attack Proxy (ZAP), NMAP, SSLyze y Selenium, todo bajo un desarrollo en Python.
Para la verificación de vulnerabilidades en arquitecturas técnicas se utiliza **Vats**, que también tiene la capacidad de analizar la seguridad en imágenes/contenedores.
Es un desarrollo realizado en Python, donde se integra **OpenSCAP** a través de agentes con un servidor central que centraliza toda la información.

Próximos pasos 'Platform Security'

Para toda la parte de **IDM / AuthN** hay varios puntos importantes donde poner el foco y empezar a tratar:

- **UMA (*User Managed Access*)**: para poder permitir que una aplicación acceda a datos sensibles de un usuario, es necesario que el usuario dé su autorización. Los procesos de autorización de acceso a los datos personales de un usuario deben ser lo más estándar posibles. UMA (https://docs.kantarainitiative.org/uma/rec-uma-core-v1_0_1.html) ofrece un ***framework*** de gestión de las autorizaciones que parece adecuado (teniendo en cuenta lo joven del mismo).
- **Autenticación adaptativa**: es un método para definir el nivel de identificación requerido para un usuario en función de su comportamiento habitual y el nivel de riesgo detectado. Para esta evaluación se necesita realizar tanto un análisis de las aplicaciones, mecanismos de autenticación disponibles y su fiabilidad, como del comportamiento del usuario. Este último punto es especialmente importante para detectar comportamientos anómalos que incrementan el riesgo de la operación. Ejemplo de empresas que implementan soluciones en este ámbito: Behaviosec, Experian.
- **Axiomatics**: es una empresa de referencia en el mundo de la autorización externalizada. Son pioneros en la evolución del estándar XACML, su adopción y su customización. En muchos sentidos marcan el paso en lo tocante a EAM (*Externalized Authorization Management*).
  - ALFA (*Abbreviated Language For Authorization*): DSL (*Domain Specific Language*), de código abierto desarrollado por Axiomatics, que simplifica la creación de políticas de autorización.
- **OpenID Connect**: OpenID Connect es el estándar hacia el que están evolucionando las soluciones de federación de identidades y delegación de la autenticación. Es un estándar de autorización construido sobre OAuth 2.0 que permite cubrir casos de uso de delegación de autenticación y autorización similares a los proporcionados por SAML.
- **Open Banking**: es un estándar que permite la interoperatividad de información de cuentas entre bancos y terceras partes. Este estándar está impulsado por el sistema bancario de Reino Unido, a pesar de lo cual se adapta a la normativa europea PSD2 (*Payment Service Providers 2*). Desde el punto de vista de la autenticación y la federación de identidades proporciona un marco de referencia para la decisión de la evolución de las piezas que se encuentran actualmente en desarrollo en este ámbito (Nauthilus, SerVal, LUX, Armadillo).
- **Forgerock - OpenIDM**: en uno de nuestros clientes se va a usar como solución de identidad global, e integrarla con el resto de productos de seguridad de la plataforma.

Para todo lo que tiene que ver con hacking ético y forense e incident response en cloud se seguirá trabajando en la línea de creación de herramientas que cumplan las necesidades actuales, apoyándose con tecnologías open source si se necesitase

Para todo lo que tiene que ver con ***hacking* ético** y **forense e *incident response* en *cloud*** se seguirá trabajando en la línea de creación de herramientas que cumplan las necesidades actuales, apoyándose con tecnologías ***open source*** si se necesitase.

En cuanto a la **criptografía**, existen diversas vertientes en las que trabajar, como mejorar lo actual e intentar ir abordando las problemáticas futuras:

- ***As a Service***: parte del futuro de este tipo de criptografía como servicio es depurar los procesos para mejorar los tiempos de respuesta y ampliar las distintas tecnologías de almacén de secretos que se ofrecen, como **HSM LunA o Servicios de HSM de proveedores como AWS**.
- **Futuro de la criptografía**: actualmente desde los Labs de BBVA Next Technologies se está trabajando en la problemática actual de la seguridad de los algoritmos criptográficos actuales y la problemática de la distribución, creación y almacenamiento de claves de forma segura. Los temas que se están trabajando y tienen un recorrido de 10 años incluyen: **criptografía postcuántica** (diseño de algoritmos frente a la computación cuántica), **criptografía homomórfica** (computación de datos cifrados), ***searchable encryption*** (búsqueda de información en datos cifrados) y ***whitebox cryptography*** (protección de claves criptográficas en entornos no confiables, como la memoria RAM o un proveedor ***cloud***).

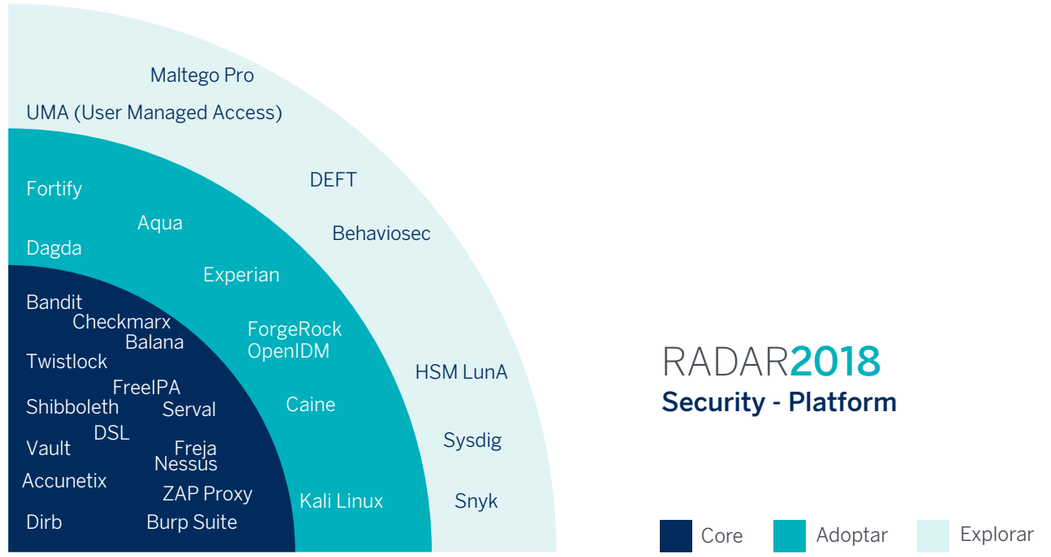
En cuanto al **almacén de secretos,** se están buscando tecnologías que den la funcionalidad de los HSM de forma ***software,*** pudiendo escalar según necesidad de los servicios -que es la principal desventaja de los HSM-, aunque lo más cercano es la integración con HSM, tanto en nube AWS o CPD.

Para toda la parte de DevSecOps, en **seguridad en código** se tratará de incluir y unificar los procesos de revisión de análisis de dependencia y la revisión estática de seguridad del código, disponibilizando los mismos bajo filosofía DevSecOps.

En **seguridad en imágenes *docker*** se buscará cerrar el ciclo de monitoreo, desde la automatización en ***pipelines*** que ya tenemos, hasta cerrar con el monitoreo continuo de lo que se está ejecutando en los diferentes ambientes.

Para **BDD Security** la evolución pasa por la automatización de nuevas pruebas de seguridad. Estas pueden cubrir un mayor espectro de análisis e integrar dicho ***framework*** con los procesos de CI/CD. También se quieren analizar herramientas ***open source***, o alternativas como **BSD Security** (en Java), o **GAUNTLT** y ver qué pueden aportar a nuestras necesidades, o qué pruebas tienen y qué valor pueden aportar para que sean añadidas en el ***framework***.

En lo referente a la verificación técnica, se pretende ampliar la integración de Vats para nutrirse de otras tecnologías como **AWS Inspector, RedHat insights, etc.** y poder contrastar datos de varias fuentes e incluso abstraer un cambio de tecnología si fuera necesario. ■



Estado actual 'Networking Security'

En uno de nuestros proyectos se ha definido el uso de una **SDN** en vez de usar el modelo anterior de redes. Como solución de SDN se ha elegido y se está usando en las dos plataformas **Cisco ACI**. En cada una de las plataformas se integra con OSP o RHEV, respectivamente.

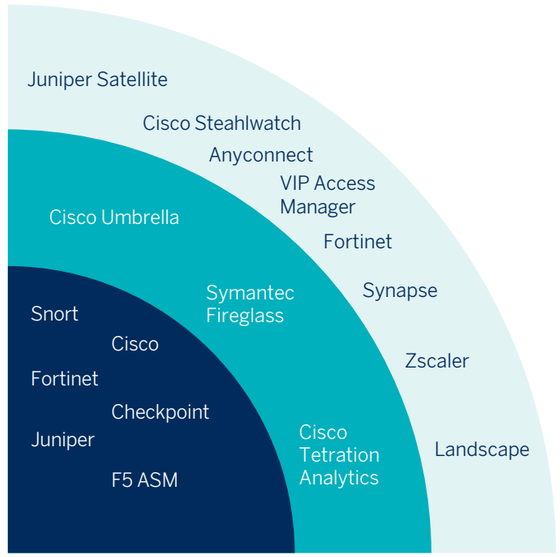
La posibilidad de tener el plano de control y datos separados, junto con las posibilidades de automatización y microsegmentación hacen avanzar un gran paso tanto en la gestión como en aumentar la seguridad global

La posibilidad de tener el plano de control y datos separados, junto con las posibilidades de automatización y microsegmentación, hacen avanzar un gran paso tanto en la gestión como en aumentar la seguridad global.
Por otro lado, en OSP se usa el concepto de ***Security Groups*** (al igual que en ***clouds*** públicas) para poder restringir a nivel L4 los puertos y protocolos que se exponen. Debido a las restricciones de OSP, sólo se puede ser reactivo en el caso en que alguien exponga un puerto. Para ciertas VMs críticas se usa **IPTables** y/o **TCP Wrappers** para securizarlos.
Adicionalmente, la segmentación de entornos se realiza con parejas de ***firewalls*** **Checkpoint** y el ***firewall*** perimetral es una pareja de los nuevos **NGFW Cisco Firepower**.
Por parte del *Network Access Control* (NAC), la solución escogida es **CISCO ISE**, junto con una solución global de Acceso Remoto. En nuestro caso, para todos los usuarios técnicos que pertenecen a la infraestructura se está usando un **concentrador VPN Fortigate**, teniendo una VPN de L4 (VPN SSL). Para poder acceder se necesita de un 2FA y existe un perfilado de los usuarios, pudiendo sólo acceder a los recursos que ha de tener acceso.
Actualmente los flujos de red se están analizando con **nfsen** y toda la información de logs y eventos de seguridad se analizan con el **SIEM Splunk**.

Actualmente los **flujos de red se están analizando con nfsen** y toda la información de **logs y eventos de seguridad se analizan con el SIEM Splunk**

Próximos pasos 'Networking Security'

Para poder tener una mejor visibilidad de la red y los flujos, y realizar una mejor segmentación, se está realizando una PoC con **Cisco Tetration Analytics**, que a grandes rasgos es una plataforma de ***big data*** para analizar los flujos de red.
Adicionalmente, el futuro proyecto de **Synapse** cambiará todo el ***backbone*** y la conexión con las redes internacionales, por lo que se realizarán nuevos cambios e iniciativas. Además se intentará tener un Network Policy para que los Security Groups se puedan gestionar de forma proactiva.
Otra parte a proteger es *antimalware* a nivel de DNS, y para ello, la tecnología elegida es Cisco Umbrella. ■



BBVA
Next Technologies

@bbvaNext company/bbva-next-technologies www.bbvanexttechnologies.com