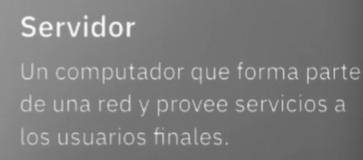
Platzi wallet.

**APP de ejemplo para explicar los conceptos que vamos a ver a lo largo del curso**:  
Billetera virtual: PlatziWallet.  
Funcionalidades:

1. Pago a personas (girar dinero y solicitar dinero).
2. Pago a terceros (pagar netflix, amazon)
3. Generar códigos QR para que lo escaneen y me puedan hacer un pago.
4. Consignación: diferentes medios por los cuales puedo agregarle dinero. Cajero electrónico, tarjeta de crédito, etc.
5. Integración para que se pueda retirar dinero.

Servidores y almacenamiento:



Ejemplo:

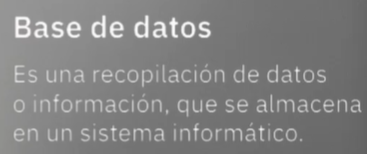
A screenshot of a phone

Description automatically generatedA screen shot of a phone

Description automatically generated

Nube: servidores.

Clase: Bases de datos.

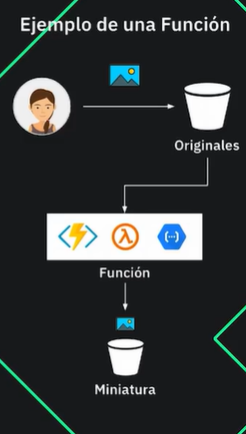
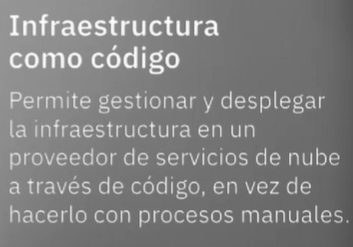
A grey background with white text

Description automatically generated



CLASE: Microservicios, funciones y contendores.

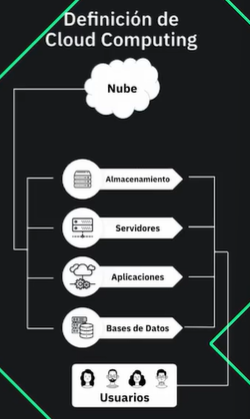
AWS (lambda functions),



CLASE: Qué es on-premises?

**On-premises** se refiere a la ubicación física de recursos informáticos, como servidores, sistemas de almacenamiento y redes, que se encuentran y operan dentro de las instalaciones de una organización

¿Qué es CLOUD computing o nube?



CLASE: ¿Por qué usar Cloud Computing o nube?

1. **Agilidad**: La capacidad de escalar y desplegar recursos rápidamente según la demanda, lo que agiliza el desarrollo y la entrega de aplicaciones y servicios.
2. **Ahorro de costos**: Reducción de los gastos de inversión en infraestructura y mantenimiento, además de pagar solo por lo que se utiliza, lo que optimiza los costos operativos.
3. **Elasticidad**: La habilidad de aumentar o disminuir los recursos de manera automática para adaptarse a picos de carga o cambios en la demanda, asegurando un rendimiento óptimo.
4. **Innovación**: Al eliminar la preocupación por la gestión de infraestructura, los equipos pueden centrarse en la creación y mejora constante de aplicaciones y servicios innovadores.
5. **Despliegue global en minutos**: La posibilidad de implementar aplicaciones y recursos en múltiples regiones del mundo de manera rápida y sencilla.
6. **Catálogo de servicios**: Acceso a una amplia variedad de servicios preconfigurados, como bases de datos, análisis y machine learning, lo que acelera el desarrollo y ahorra tiempo.

CLASE: ¿Por qué una arquitectura en Cloud Computing o nube es diferente?

1. **Modelo de costos (CAPEX a OPEX)**: Cambio del gasto de capital (CAPEX) tradicional en infraestructura a gastos operativos (OPEX) variables según el uso, lo que permite una mayor flexibilidad financiera.  
   **2. Modelo de entrega/despliegue (Automatización y creación de ambientes)**: Automatización de la implementación y creación de ambientes a través de código, lo que agiliza la configuración y gestión de recursos.  
   **3. Patrones nativos de nube**: Diseño de aplicaciones específicamente para aprovechar las características únicas de la nube, como escalabilidad horizontal y alta disponibilidad.  
   **4. Escalabilidad y resiliencia**: Capacidad de aumentar dinámicamente los recursos y diseñar sistemas para resistir fallas y mantener la disponibilidad en diferentes zonas y regiones.  
   **5. Tecnologías disruptivas**: Integración de tecnologías disruptivas como Inteligencia Artificial (IA), análisis de datos, robótica, satélites y automatización para innovar en soluciones.  
   **6. Modelo de seguridad compartida**: Distribución de la responsabilidad de la seguridad entre el proveedor de servicios en la nube y el usuario, asegurando una implementación segura y cumplimiento normativo.

Ambiente pruebas, desarrollo y producción.

Clase 11: Nuber privada, publica, híbrida, multinube.

1. NUBE PRIVADA: solamente puede ser accedida por nosotros como empresa. Se puede pensar como un ON-PREMISE. Con nuestro datacenter, servidores, almacenamiento e infraestructura. Para instalar APP y DB, usar el STORAGE, etc. La empresa es la única que tiene acceso a esa nube.
2. NUBE PÚBLICA (AWS, AZURE, GCP, etc): donde múltiples empresas tienen acceso. Son los CLOUD PROVIDERS del mercado.  
   NUBE HÍBRIDA: sistema o aplicación que funciona con una nube pública coexistiendo con otra privada. Ej: PlatziWallet la desplegamos en AWS sin embargo la aplicación todavía tiene una DB que no hemos migrado hacia la nube.
3. MULTI NUBE: cuando coexisten 2 nubes públicas. Ej: PlatziWallet la tenemos desplegada en AWS, sin embargo toda la parte de autenticación de usuarios la tenemos en un AZURE en un servicio de directorio activo. Necesitamos conectividad entre AWS y AZURE para hacer la autenticación de usuarios.
4. MULTI NUBE HÍBRIDA: 2 nubes públicas + 1 nube privada. Ej: PlatziWallet corre en AWS, pero tenemos la parte de procesamiento de datos y analítica en GCP. Todos los eventos que genera PlatziWallet los enviamos a GCP. Pero la APP tiene que consumir una DB que está en on-premise en una nube privada.

CLASE: Ejemplo de arquitectura Cloud Native.

Características Arquitectura:  
Altamente disponible (AZ): 2 zonas de disponibilidad.

1. CAPA DE CONECTIVIDAD: para el on-premises, las instalaciones de la empresa o una capa de networking. A través de un VPN o una conexión dedicada que nos ofrece Cloud P.
2. ZONA PRIVADA: donde corre el backend de nuestra aplicación. No va a estar expuesta a internet. Nosotros exponemos nuestros servicios, pero nuestra aplicación, los contenedores, las funciones quedan en esta zona privada.
3. ZONA PÚBLICA: servicios que deben estar expuestos hacia internet. Reciben los requests directamente de los usuarios y se comunican con la capa privada donde está el backend.  
   DNS: nombre de dominio (platziwallet com). Le agregamos las reglas de enrutamiento. Se pueden crear a nivel de los DNS opciones de redireccionamiento (basados en el nombre de dominio).  
   CDN (content delivery network ): expone el servicio de forma global a través del uso de ubicaciones de borde. Ubicaciones de borde son datacenter pequeños distribuidos por el mundo sobre los cuales puedo poner el contenido estático o dinámico de la aplicación. Esto ayuda a que la experiencia de usuario sea mucho mejor.  
   WAT + CERTIFICADO ( seguridad). Va a ver usuarios que van a estudiar cómo toman ventaja para hacer fraude. Web Application Firework + Certificado de seguridad para agregarle reglas de seguridad a nuestra aplicación. De esta manera protegemos nuestra aplicación de ataques de denegación de servicios, sql injections, ataques de script  
   API GATEWAY: recibe toda la información se la manda al BALANCEADOR de aplicaciones. Detrás del balanceador podemos tener nuestro backend corriendo en Kubernetes. Este clúster de Kubernetes donde está corriendo PlatziWallet y donde están los microservicios de pagos, cobro, recarga, login, biometría, etc. Estos tienen una capa de almacenamiento (objetos, bloques y archivos).  
   DESPLIEGUE: para desplegar nuestro backend tenemos: Kubernetes, argo y helm

A screenshot of a phone

Description automatically generated

Clase: ¿IaaS, PaaS y SaaS?

IaaS (Infraestructura como Servicio), PaaS (Plataforma como Servicio) y SaaS (Software como Servicio) son tres modelos de servicio en la nube que ofrecen diferentes niveles de control y responsabilidad a los usuarios sobre la infraestructura, la plataforma y el software. Aquí tienes una descripción breve de cada uno:

* 1. IaaS (Infraestructura como Servicio): En IaaS, los proveedores de la nube ofrecen infraestructura virtualizada, como servidores, almacenamiento y redes. Los usuarios pueden crear, configurar y administrar máquinas virtuales y recursos según sus necesidades. Esto brinda un alto nivel de control sobre la configuración y el sistema operativo, pero los usuarios también son responsables de administrar y mantener el software y las aplicaciones en estas máquinas virtuales.  
     Ejemplo: Amazon EC2, Microsoft Azure Virtual Machines.
  2. PaaS (Plataforma como Servicio): En PaaS, los proveedores ofrecen un entorno de desarrollo y ejecución completo para que los desarrolladores creen, prueben y desplieguen aplicaciones sin preocuparse por la infraestructura subyacente. PaaS proporciona herramientas y servicios para simplificar el proceso de desarrollo y permite centrarse en la codificación y la innovación, mientras el proveedor maneja la administración de servidores y sistemas.  
     Ejemplo: Google App Engine, Heroku.
  3. **SaaS (Software como Servicio)**: En SaaS, los usuarios acceden a aplicaciones y software a través de la nube sin la necesidad de instalar ni mantener el software localmente. Las aplicaciones están completamente gestionadas por el proveedor de la nube y se entregan a través de un navegador web.  
     Ejemplo: Google Workspace, Microsoft 365, Salesforce.