

# Organización de los sistemas de Azure



## Componentes.

- Arturo Cortés Sánchez
- Abel José Sánchez Alba
- Hamada Bouhacida
- Khaled Mohamed Elamine Saidani

### **PSI 7.1 Identificación de las necesidades de Infraestructura Tecnológica**

Es complicado acotar las necesidades tecnológicas de un sistema tan grande como lo es Azure, no obstante basándonos en datos históricos podemos realizar algunas aproximaciones. Según esta

<https://www.quora.com/How-many-servers-has-Microsoft-Azure-worldwide>fuente,

Azure en 2017 tenía alrededor de 3 millones de servidores. Esta otra fuente afirma que en 2017 Azure tenía 36 regiones geográficas

<https://azurespain.azurewebsites.net/2017/10/11/donde-estan-mis-servicios-regiones-emparejados-y-zonas-de-disponibilidad/>, que básicamente son agrupaciones de datacenters.

Cada una de las regiones geográficas es capaz de ofrecer solo ciertos servicios, por lo que podemos suponer que los servidores están especializados en ciertas tareas. De forma un tanto inocente vamos a asumir que un cuarto de los servidores están especializados en procesamiento de datos mediante CPU, otro cuarto en labores de inteligencia artificial u otras tareas que requieran GPU, otro cuarto en tareas que requieran grandes cantidades de memoria principal y el último cuarto en almacenamiento de datos.

Dentro del mundo de las CPUs hay gran cantidad de proveedores, como las CPUs POWER, SPARC o ARM, pero los fabricantes principales son Intel y AMD con su arquitectura X86.

En el apartado de los sistemas operativos tenemos Linux, Windows server, macos server o FreeBSD. El uso de los dos últimos es bastante residual, por lo que a la hora de la verdad solo hay dos alternativas.

Actualmente solo hay dos proveedores de GPUs, Nvidia, con su gama tesla y AMD con su gama Instinct.

Para el caso del almacenamiento podemos elegir entre un gran abanico de fabricantes, como Western Digital, Seagate, Kingston, Toshiba etc. Pero mas aun importante que elegir fabricante es elegir la tecnología, discos duros mecánicos o discos de estado sólido.

### **PSI 7.2 Selección de la Arquitectura Tecnológica**

Para la CPU la opción más lógica es decantarse por la arquitectura x86, ya que es la única arquitectura capaz de ejecutar Windows server, y siendo Azure una división de Microsoft, es lógico querer ejecutar dicho sistema en los servidores. Dentro de la arquitectura x86 hay únicamente dos proveedores, Intel y AMD. A fecha de escritura de este documento los procesadores AMD son mejor opción que los de Intel, pero son considerablemente más difíciles de obtener. Así que lo más óptimo es utilizar procesadores AMD en la medida de lo posible y rellenar con procesadores Intel.

Para los sistemas operativos lo mejor sería usar Linux ya que es el sistema dominante en el mundo de los servidores, pero dado que Azure pertenece a Microsoft, es lógico querer vender licencias de Windows server. MacOS server solo es posible usarlo en sistemas Apple, y a pesar de su nombre es más bien una extensión del macos tradicional para añadir herramientas típicas de servidores. FreeBSD podría ser una buena alternativa, pero al ser considerablemente menos usado que Windows o Linux, su soporte está más limitado. Una vez más lo óptimo es una solución híbrida, ofrecer a los usuarios Windows server y Linux.

En el mundo de las GPUs estamos más limitados, ya que las únicas tarjetas gráficas actuales para servidores son la gama Tesla de Nvidia y la gama Instinct de AMD. La mayoría de frameworks de desarrollo de inteligencia artificial están basados en la tecnología Cuda de Nvidia. Por lo que lo lógico sería utilizar GPUs Nvidia Tesla.

Para los servidores de almacenamiento de datos habría que usar tanto discos duros tradicionales como SSDs, ya que la capacidad de almacenamiento que ofrecen es muy superior a la de los SSDs, pero a su vez ofrecen latencias demasiado altas y velocidades de lectura demasiado lentas para según que sistemas. Es por eso que también son necesarios los SSDs, para cubrir las carencias de los discos duros mecánicos.

## Estudio de viabilidad del sistema

Durante esta sección se realizará un análisis de las necesidades a la hora de diseñar una solución inicial. Este análisis debe tener en cuenta las posibles restricciones, y una vez realizado se explorarán las posibles alternativas a la solución propuesta.

### **EVS 1.1 Estudio de la Solicitud**

La principal necesidad de un sistema como Azure es una red robusta de datacenters. Construir un datacenter no es una tarea sencilla, ya que además de todo el equipamiento tecnológico necesario hace falta una cantidad considerable de terreno así como acceso a electricidad barata y refrigeración. Es por esto que los sitios más idóneos suelen ser lugares aislados a latitudes relativamente altas. No obstante esto supone el problema de estar alejado de núcleos de población.

Objetivos:

- Ver si el servicio tiene suficiente demanda
- Identificar el coste aproximado del sistema
- Identificar costos ocultos así como de mantenimiento

## **EVS 1.2 Identificación del Alcance del Sistema**

Dado que Azure provee una gran cantidad de servicios tecnológicos de los que otras empresas hacen uso para construir los suyos propios, Azure debe ser capaz de proveer un conjunto de utilidades suficientemente amplio para cubrir estas necesidades. Un problema comúnmente asociado a servicios de gran complejidad como este es su curva de aprendizaje así como su dificultad de uso, por lo que para tener cierto éxito es necesario cuidar como se muestra la información a los usuario así como la documentación y las APIs.

## **EVS 1.3 Especificación del Alcance del EVS**

A la hora de hacer el estudio de viabilidad se realizarán las siguientes tareas:

- Exploración del mercado de proveedores cloud y otros servicios similares.
- Identificación de posibles lugares para la construcción de datacenters.
- Comparación del hardware de servidores disponible en el mercado.
- Identificación del software necesario para construir el sistema.
  - Ver cual de este software ya existe y está disponible para su uso.
  - Ver si hay software comercial que se pueda usar o si hay que programarlo desde cero.

Hamada

## **EVS 2.1 Valoración del Estudio de la Situación Actual**

La situación actual del sistema azure desde la perspectiva de la administración de sistemas y en según el sistema a desarrollar es el siguiente:

- Elementos físicos de los sistemas actuales:

Microsoft Azure production network (Azure network)

Microsoft corporate network (corpnet)

- Usuarios participantes en el sistema actual:

- Administrador de servidores.

- Administrador de la base de datos.

- Administrador de redes.

## **EVS 2.2 Identificación de los Usuarios Participantes en el Estudio de la Situación Actual**

Los empleados (o contratistas) de Microsoft se consideran usuarios internos.

Azure usa identificadores únicos para autenticar a los usuarios y clientes de la organización (o procesos que actúan en nombre de los usuarios de la organización). Esto se aplica a todos los activos y dispositivos que forman parte del entorno de Azure.

- Administrador de servidores.

- Acceso a todas las instalaciones del complejo y a armarios de red. Permiso sobre el contenido y administración de las aplicaciones del sistema.

- Administrador de la base de datos.

- Acceso al CPD y al contenido y administración de las bases de datos de las aplicaciones del sistema.
- Administrador de redes.
- Acceso a todas las instalaciones del complejo y a los armarios de red.
- Empleados.

### **EVS 2.3 Descripción de los SI Existentes**

La definición del SI implica la especificación de todos los elementos del entorno tecnológico así como de las características de los usuarios implicados en él, Productos de Azure :

Administración y Gobernanza.

Almacenamiento.

Análisis.

Bases de datos.

Blockchain.

Contenedores.

DevOps.

Herramientas para desarrolladores.

### **EVS 2.4 Realización del Diagnóstico de la Situación Actual**

Podemos utilizar la supervisión para obtener información detallada sobre cómo funciona el sistema. La supervisión es una parte importante para mantener los objetivos de calidad del servicio. La recopilación de datos de seguimiento se puede utilizar en las siguientes situaciones:

- Asegúrese de que el sistema esté en buenas condiciones.

Monitorear la disponibilidad del sistema y sus componentes.

Asegúrese de que el procesamiento del sistema no se deteriore inesperadamente a medida que aumenta la carga de trabajo.

Asegúrese de que el sistema cumpla con los acuerdos de nivel de servicio (SLA) con los clientes.

Proteja la privacidad y seguridad del sistema, los usuarios y sus datos.

Seguimiento de las operaciones realizadas como revisión a efectos organizativos.

Uso diario del sistema e identificación de tendencias que pueden generar problemas para actuar con anticipación.

### **EVS 3.1 Identificación De las Directrices Técnicas y de Gestión**

Con este fin, se recoge información sobre los estándares y procedimientos que deben considerarse al proponer una solución, relativos a:

Políticas técnicas:

Gestión de Proyectos (seguimiento, revisión y aprobación final).

Desarrollo de Sistemas (existencia de normativas, metodologías y técnicas de programación).

Arquitectura de Sistemas (centralizada, distribuida).

Política de Seguridad (control de accesos, integridad de datos, disponibilidad de aplicaciones).

Directrices de Planificación.

Directrices de Gestión de Cambios.

Directrices de Gestión de Calidad

### **EVS 3.2 Identificación De Requisitos**

Dado que Azure será un servicio en la nube capaz de almacenar datos y aplicaciones integradas, es esencial disponer de un sistema hardware resistente y flexible, por lo que esto será el requisito fundamental, una red de dispositivos aptos para virtualización, conexiones eléctricas e internet, servidores, sistemas de almacenamiento de información de formas interna y sistemas para almacenar la información de los propios usuarios, es decir, un sistema donde cada usuario tenga su espacio de almacenamiento web y despliegue. Así como disponer de una API para los usuarios.

Además, una vez se tenga la infraestructura como servicio, será posible ofrecer software como servicio mediante la creación de herramientas integradas dentro de la propia estructura, así un usuario podrá emplear el almacenamiento y las tecnologías que se ofrecen, por lo que no necesitará salir del entorno de azure.

### **EVS 3.3 Catalogación De Requisitos**

Los requisitos se catalogarán como :

1. Hardware orientado a redes
2. Hardware orientado a virtualización
3. Hardware orientado a almacenamiento y bases de datos
4. Desarrollo de API
5. Software as a Service

### **EVS 4.1 Preselección de Alternativas de Solución**

Las alternativas a la solución son escasas, debido a que el proyecto es crear un ambiente donde el cliente pueda tener un espacio de almacenamiento y servicios software derivados de este, sin embargo si existen alternativas en cuanto a la implementación.

1. Subcontratación de la red y hardware
2. Ofertar servicio a otras empresas con opción a compra de sus softwares.
3. Subcontratación del hardware de almacenamiento y bases de datos.

### **EVS 4.2 Descripción de las Alternativas de Solución**

Una alternativa es la subcontratación del desarrollo del sistema hardware y de red, así, una de las mayores fuentes de problemas terminales estaría eliminada. Otra opción es ofrecer el servicio de almacenamiento y virtualización a start-ups para su desarrollo con opción a compra de sus servicios, así se ampliaría la oferta de servicios de la empresa.

Por último, siguiendo la filosofía de la primera alternativa, los servidores de bases de datos también son muy susceptibles a fallos, especialmente con el uso se que se plantea dar. Por ello se propone la subcontratación de este servicio, para así poder tener un servicio de almacenamiento fiable y centrar esfuerzos en el modelo de negocio de SaaS.

### **EVS 5.1 Estudio de la Inversión**

La alternativa a la solución de la subcontratación de servicios podría hacer que el precio de servicio al usuario final aumente y por tanto no ser rentable o competitivo en el mercado.

La oferta de servicios a empresas con opción a compra o renta de parte o el total de la empresa o sus softwares es una alternativa perfectamente viable que encaja con el modelo de microservicios que ofrece microsoft. Además de fomentar así la entrada al sistema de usuarios y seguidores de la plataforma comprada.

La subcontratación del hardware de almacenamiento y bases de datos presenta el mismo problema que la primera alternativa pero aplicada a la parte de almacenamiento del sistema. Cuantos más intermediarios se interpongan en estas secciones del sistema, más subirá el precio y por tanto más puede afectar al precio final y su competitividad.

### **EVS 5.2 Estudio de los Riesgos**

Las detecciones de riesgo en Azure AD Identity Protección incluyen todas las acciones sospechosas identificadas, relacionadas con las cuentas de usuario en el directorio.

**Identity Protection** proporciona a las organizaciones el acceso a recursos eficaces, para ver y responder rápidamente a estas acciones sospechosas.

Hay dos tipos de riesgo, Usuario e Inicio de sesión y dos tipos de detección o cálculo Tiempo real y Sin conexión.

Es posible que las detecciones en tiempo real no se muestren en los informes durante un tiempo comprendido entre cinco y diez minutos. Es posible que las detecciones sin conexión no se muestren en los informes durante un tiempo comprendido entre dos y veinticuatro horas.

Un riesgo de usuario representa la probabilidad de que una identidad o cuenta determinada esté en peligro.

Estos riesgos se calculan sin conexión, usando orígenes de inteligencia sobre amenazas internas y externas de Microsoft, incluidos los investigadores de seguridad, los profesionales de la aplicación de la ley, los equipos de seguridad de Microsoft y otros orígenes de confianza.

### **EVS 6.1 Convocatoria de Presentación**

En este paso se realiza una convocatoria para la presentación del sistema al Comité de Dirección y se espera la confirmación por parte del Comité de Dirección. Se debe preparar una presentación en la que se vea bien reflejado el estado del sistema. Entre los apartados que se tomarán habrá un informe detallado con la evaluación obtenida de las pruebas realizadas y el plan de mantenimiento, mostrando todos los procesos que se realizaron y que se han probado correctamente.

### **EVS 6.2 Evaluación de Alternativas y Selección**

Finalmente, de acuerdo a la arquitectura tecnológica propuesta y los requisitos identificados

anteriormente se proponen los siguientes elementos del sistema:

- Matriz Procesos / Localización geográfica.
- Las nuevas aplicaciones con las que contará el sistema se desplegarán en un servidor en el CPD de la empresa, ubicado en el sótano del edificio.
- Matriz Datos / Localización geográfica.- Los datos almacenados y manejados por las aplicaciones del sistema se guardarán en volúmenes guardados en discos SSD.
- Entorno tecnológico y comunicaciones.
- Estrategia de implantación global del sistema. Ante el entorno tecnológico planteado y los requisitos descritos para el sistema se propone la siguiente estrategia de implantación para la evaluación de su viabilidad.
- Instalación en un armario del CPD de tres nuevos blades, uno para cada nueva aplicación que incluirá el sistema (control de acceso, administración de tareas y fichaje).
- Instalación de otro armario del CPD de los discos necesarios para almacenar los datos de dichas aplicaciones y conectar los armarios. Con instalaciones en armarios separados se mejora la organización y se minimizan errores en la administración del sistema.
- Instalación de un switch de acceso red en cada planta para la administración del acceso a red.
- Extensión de la red LAN existente para cubrir las necesidades de instalación de los puntos de acceso inalámbricos y de los terminales de control de acceso biométrico.
- Instalación de nuevos punto de acceso inalámbricos. Al menos uno por cada área de acceso restringido o por cada 100m2 aproximadamente. Cifras más concretas se darán cuando se realice un análisis del sistema.
- Instalación de los terminales de control de acceso con perno (junto con las cerraduras de llave maestra) en las puertas de entradas y salidas de áreas de acceso restringido.
- Adhesión de todos los lectores de huellas simples a los terminales tipo Tablet de uso para empleados.
- Distribución de todos los terminales tipo tablet a todas las instalaciones donde sean requeridos. Habilitación de un puerto de carga por cada terminal tipo tablet en un emplazamiento cercano a su lugar de uso común.
- Registro en el sistema de todos los dispositivos que tiene permitido conectarse a la red de la organización.

Dada la situación económica de la empresa y las inyecciones de dinero asiduas que recibe la organización se concluye que tiene los recursos necesario

### **EVS 6.3 Aprobación de la Solución**



Se presenta el sistema al Comité de Dirección según el plan previsto y se aprueba formalmente el sistema.

Usaremos una aprobación de la versión en Azure Pipelines como ayuda para coordinar los cambios de esquema de la base de datos entre desarrolladores y administradores de bases de datos.

Objetivos de este módulo:

Conectar una base de datos relacional a nuestro sitio web usando Azure SQL Database y App Service como tecnologías de ejemplo

Coordinar los cambios en el esquema de la base de datos mediante una aprobación de la versión

Usar PowerShell para compartir variables a lo largo de las fases de canalización.

### **ASI 1.1 Determinación del Alcance del Sistema**

Vamos a expandir en los requisitos obtenidos durante la EVS

- Hardware orientado a redes: aparatos informáticos, que facilita el uso de una red informática. Típicamente, esto incluye enrutadores, switches, hubs, gateways, puntos de acceso, tarjetas de interfaz de red etc
- Hardware orientado a virtualización: es un enfoque de virtualización de plataforma que permite una virtualización completa eficiente utilizando la ayuda de las capacidades del hardware, principalmente de los procesadores del host
- Hardware orientado a almacenamiento y bases de datos: Discos de estado sólido para aplicaciones que requieran baja latencia y discos duros tradicionales para almacenamiento sin este requisito. Opcionalmente se pueden utilizar cintas magnéticas.
- Desarrollo de API: interfaz que define las interacciones entre varias aplicaciones de software. Define los tipos de llamadas o peticiones que se pueden hacer, cómo hacerlas, los formatos de datos que se deben utilizar, las convenciones a seguir, etc.
- Software as a Service: Modelo de licencia de software en el que el software se licencia por suscripción y se aloja de forma centralizada. A veces se denomina "software a la carta". Las aplicaciones SaaS también se conocen como software bajo demanda y software basado en la web/alojado en la web.

Glosario:

- Datacenter: Espacio donde se concentran los recursos necesarios para el procesamiento de la información de una organización. Dichos recursos

consisten esencialmente en unas dependencias debidamente acondicionadas, computadoras y redes de comunicaciones.

- Servidor: Hardware o software que proporciona funcionalidad a otros programas o dispositivos, llamados clientes. Los servidores pueden proporcionar varias funcionalidades, a menudo llamadas "servicios".
- Servicio: Funcionalidad de software o a un conjunto de funcionalidades de software (como la recuperación de información especificada o la ejecución de un conjunto de operaciones) con una finalidad que diferentes clientes pueden reutilizar para diferentes propósitos
- Máquina virtual: Proporciona una plataforma de sistema completa y soporta la ejecución de un sistema operativo completo. Suelen emular una arquitectura existente y se construyen con el propósito de tener múltiples instancias de máquinas virtuales que conduzcan a un uso más eficiente de los recursos informáticos, tanto en términos de consumo de energía como de rentabilidad.
- Aplicación: programa informático diseñado para llevar a cabo una tarea específica que no esté relacionada con el funcionamiento del propio ordenador, normalmente para ser utilizado por los usuarios finales.
- Usuario: Persona que hace uso de los servicios de Azure, ya sea de forma directa, por ejemplo programando para la API o de forma indirecta a través de la aplicación de un cliente.

### **ASI 1.2 Identificación del Entorno Tecnológico**

Dada la gran cantidad de usuarios que los grandes proveedores de cloud, los requisitos tecnológicos son considerables. Como se comentó anteriormente, Azure tiene más de tres millones de servidores divididos en 36 regiones geográficas, lo que nos da más de 80000 servidores por región. No todas las regiones son capaces de proveer los mismos servicios, pero está claro que cada uno de esos servidores necesita al menos una CPU, memoria RAM y almacenamiento permanente. A esto hay que sumarle el resto del hardware, por ejemplo la placa base, el sistema de red, refrigeración, alimentación etc. Aunque dependiendo del diseño de los servidores estos últimos componentes pueden ser en algunos casos compartidos, suponiendo así cierto ahorro.

### **ASI 1.3 Especificación de Estándares y Normas**

A la hora de poner en marcha un servicio como este hay cantidad de leyes a tener en cuenta. Algunas de las mas relevantes son la leyes de protección de datos, especialmente la europea. La RGPD exige a las empresas que protejan los datos personales y la privacidad de las transacciones que se realizan en los Estados miembros de la UE. Protege la información básica de identidad, como el nombre, la dirección y los números de contacto; los datos de la web, como la dirección IP y los datos de las cookies; la salud, los datos genéticos, los datos biométricos y la orientación sexual; los datos raciales y étnicos; y las opiniones políticas.

Otras leyes importantes son por ejemplo las relacionadas con el consumo energetico. En la actualidad la unión europea está preparando una ley para vigilar el consumo de los centros de datos, que al ser un sector creciente cada vez es mas alto

#### **ASI 1.4 Identificación de Usuarios Participantes y Finales**

En los sistemas cloud como azure hay tres tipos de usuarios:

- Usuarios administradores: Este tipo de usuarios trabaja para la empresa y se encarga del mantenimiento y/o desarrollo del sistema. Su numero es mas bien reducido.
- Clientes directos: Estos usuarios hacen uso de los servicios cloud de la empresa para desplegar sus propios servicios. Son los proveedores principales de ingresos para la empresa y si bien su número es bastante mayor que el de los administradores, palidece en comparación con los clientes indirectos.
- Clientes indirectos: Clientes de los clientes directos. Estos usuarios hacen uso de los servicios que los clientes directos despliegan utilizando los servicios de la empresa. Son los mas numerosos con diferencia.

#### **ASI 2.1 Obtención de Requisitos**

Los requisitos son los siguientes:

- Funcionales: El sistema debe de proveer a los programadores con software o APIs que resulten familiares
- Rendimiento: El sistema debe de ser capaz de soportar gran cantidad de usuarios sin que el rendimiento se vea afectado. Además debe de proveer con hardware especializado para HPC y procesamiento altamente paralelo (GPUs) para los usuarios que lo necesiten.
- Seguridad: Dada la cantidad de usuarios que van a construir sus servicios sobre este sistema es necesario que el software esté actualizado con los últimos parches de seguridad, que todo usuario con permisos de administración mantenga estrictamente unas buenas prácticas y que el acceso físico al datacenter esté continuamente vigilado.
- Disponibilidad del sistema: La disponibilidad global del sistema debe de de ser lo mas alta posible, no obstante debe de ser posible que los usuarios puedan contratar menores tasas de disponibilidad por un precio inferior.

#### **ASI 2.2 Especificación de Casos de Uso**

Casos de uso:

- Un usuario cliente principal debe de ser capaz de contratar cualquiera de los múltiples servicios ofrecidos al público. Para ello debe de registrarse, incluir un método de pago y seleccionar y configurar el servicio que desea contratar.
- Un usuario debe de ser capaz de modificar la tarifa del servicio que tiene contratada.
- Un usuario debe de ser capaz de cancelar el servicio que tiene contratado.
- Un usuario debe de ser capaz de interactuar con las APIs de los servicios que tiene contratados. Estas APIs permiten por ejemplo que el usuario despliegue su propio servicio. Una vez que el usuario ha desplegado su servicio, este debe ser accesible desde fuera.

### **ASI 2.3 Análisis de Requisito**

Respecto a los requisitos funcionales, como es natural, los usuarios tienden a preferir sistemas con los que ya están familiarizados. En caso de que no estén familiarizados con el sistema es vital tener una buena documentación y conjunto de tutoriales para que la curva de aprendizaje sea suave.

En los requisitos de rendimiento lo mas vital es que el sistema no sufra ningún tipo de sobrecarga que pueda desembocar en que el usuario perciba lentitud o directamente una caída. El resto de requisitos están mas bien ligados a servicios específicos.

En el caso de los requisitos de seguridad y disponibilidad, la situación es similar. Los clientes que van a construir su plataforma sobre esta, por tanto esperan que el sistema subyacente sea "invisible", en el sentido de que no de problemas, ya sean de seguridad o de disponibilidad.

### **ASI 3.1 Determinación de Subsistemas de Análisis**

En la descomposición en subsistemas podemos asumir dos cosas distintas, que cada servidor es un subsistema o que cada datacenter es uno. Asumir lo segundo puede ser buena idea porque los datacenters, a diferencia de los servidores, son entidades mas o menos homogéneas

Cada datacenter estará en una ubicación determinada para ser capaz de proveer servicios con baja latencia a toda una región. Los servicios que ofrezca cada datacenter pueden variar, pues no todos los datacenters tienen por que tener disponibles el mismo hardware. No obstante todos los datacenters deben de ser capaces de proveer unos servicios básicos, como por ejemplo máquinas virtuales genéricas o despliegue de contenedores.

### **ASI 3.2 Integración de Subsistemas de Análisis**

Dado que hemos establecido que cada datacenter es un subsistema, la duplicidad conceptual es prácticamente nula porque cada entidad es lo suficientemente grande para funcionar de forma independiente, además la visión global del sistema se reduce a un conjunto de centros de datos distribuidos por todo el planeta.

#### **ASI 4.1 Identificación de Clases Asociadas a un Caso de Uso**

Clases que describen el sistema:

- Usuario
- RecursoHardware
  - CPU
  - GPU
  - Almacenamiento
- Servicio
  - Maquina virtual
  - PaaS
  - Base de datos
  - FaaS
- Las clases de la interfaz que dependerán de los diseñadores de la UI

#### **ASI 4.2 Descripción de la Interacción de Objetos**

Los objetos usuarios tienen asociados una serie de objetos servicio que han contratado.

Cada objeto servicio, además de sus propios datos, como precio o descripción, contiene además una lista de recursos hardware asociado, de forma que cada servicio sabe que hardware usar.

Los objetos RecursoHardware no están enlazados a una instancia física de hardware ya que esto dificultará la ejecución dinámica de trabajos en distintos nodos. Representan un tipo de hardware, ya sea un modelo concreto o una familia de modelos.

Los objetos como máquina virtual o cpu heredan de la clase madre que en estos casos son RecursoHardware y Servicio respectivamente.

#### **ASI 5.1 Identificación de Responsabilidades y Atributos**

Las clases contienen los siguientes atributos:

- Usuario:

- Datos del usuario
- Lista de servicios contratados
- RecursoHardware
  - Nombre
  - Cada tipo de hardware hereda de esta clase
- Servicio
  - Nombre
  - Descripción
  - Precio
  - Cada tipo de servicio hereda de esta clase

## **ASI 5.2 Identificación de Asociaciones y Agregaciones**

Las clases se relacionan de la siguiente forma:

- La clase Usuario puede contener entre cero y varios servicios. Se trata de una relación de agregación
- La clase Servicio contiene varios RecursoHardware. Se trata de una relación de
- composición.

## **ASI 5.3 Identificación de Generalizaciones**

Las relaciones de herencia entre clases son las siguientes:

- Las clases CPU, GPU, Almacenamiento y demás clases que representan un componente hardware concreto heredan de RecursoHardware.
- Las clases Máquina virtual, PaaS, Base de datos, FaaS y otras clases que representan un tipo de servicio ofertado heredan de la clase Servicio.

## **ASI 6.1 Elaboración del Modelo Conceptual de Datos**

Un posible modelo conceptual de datos sería el siguiente

Datacenter:

- Localización
- Empleados
- Servicios disponibles
- Lista de servidores

Empleado:

- Nombre
- Fecha de nacimiento
- ID
- Puesto
- Horario
- Sueldo
- Estado

Servidor:

- IP local
- Lista de recursos hardware
- Servicios disponibles
- Estado

### **ASI 6.2 Elaboración del Modelo Lógico de Datos**

Del modelo conceptual anterior podemos deducir que un la entidad datacenter está relacionada con la entidad servidor, pues un datacenter contiene varios servidores. De igual manera la entidad empleado también está relacionada con la entidad datacenter, pues varios empleados trabajan en un datacenter.

El caso de la entidad cliente es mas particular ya que los clientes no son clientes de un servidor o un datacenter, si no del servicio global. Este servicio agrupa todos los datacenters y es el que los usuarios finales ven.

Es probable que sea necesaria la creación de una entidad extra para agrupar todos los datacenters.

### **ASI 6.3 Normalización del Modelo Lógico de Datos**

El modelo de datos descrito anteriormente tiene el problema de que la entidad datacenter tiene atributos diseñados para contener directamente una lista de otras entidades, esto va en contra de la primera forma normal. Para corregir este problema se podrían sustituir dichos atributos por una relación de agregación de uno a muchos.

### **ASI 6.4 Especificación de Necesidades de Migración de Datos y Carga Inicial**

Dado que el desarrollo del sistema se está planteando desde cero, no hay necesidad de migración de datos como tal, ya que serán los clientes los que suban los datos al sistema.

La carga inicial del sistema es una carga comparativamente escasa ya que se trata de un sistema diseñado para soportar la ejecución de gran cantidad de servicios de forma simultanea, sin embargo durante el primer arranque del sistema no existe ninguno de estos servicios. En un momento así las bases de datos solo contienen la información relativa a los empleados y al propio sistema, por lo que el uso del almacenamiento es mas bien escaso.

### **ASI 7.1 Obtención del Modelo de Procesos del Sistema**

Teniendo en cuenta que anteriormente se concluyo que cada datacenter de la red global constituye un subsistema podemos definir la siguiente lista de procesos:

- Ejecución de los servicios de los clientes
  - Ejecución de maquinas virtuales
  - Ejecución de contenedores
  - Ejecución de funciones
- Tareas de seguridad y mantenimiento
  - Actualización del software
  - Comprobaciones de seguridad
- Reparación de componentes defectuosos

### **ASI 7.2 Especificación de Interfaces con otros Sistemas**

La principal interfaz de la que este sistema hace uso es Internet, ya que al tratarse de un servicio cloud, todos los clientes tienen que acceder a través de el.

Una segunda interfaz también muy importante y relacionada sería la red interna de cada datacenter, que permite a los distintos servidores comunicarse entre si de forma muy rápida. Esto permite entre otras cosas formar clústeres de procesamiento altamente paralelo.

Otra interfaz menos usada es el puerto serie que tienen los servidores, esto permite a un trabajador conectarse con un cable directamente al servidor sin necesidad de una pantalla. Esta interfaz es útil para que los sysadmins puedan solucionar posibles problemas de forma rápida.

### **ASI 8.1 Especificación de Principios Generales de la Interfaz**

Los servicios cloud cada vez proveen un catálogo de servicios mayor. Estos servicios pueden ser completamente distintos entre si, lo cual dificulta que los clientes se familiaricen con ellos. Es por esto que es necesario que las interfaces sean claras y sencillas en la medida de lo posible.



Siguiendo en la línea de lo expresado anteriormente la interfaz de usuario estará disponible únicamente a través de una página web, no obstante los usuarios que lo deseen podrán interactuar también con el sistema mediante el uso de una API.

### **ASI 8.2 Identificación de Perfiles y Diálogos**

La interfaz web del sistema deberá contar con los siguientes formularios:

- Interfaz de registro
- Interfaz de inicio de sesión
- Interfaz del perfil del usuario
  - Interfaz de métodos de pago
- Interfaz de servicios
  - Interfaz de selección y contratación de servicios
  - Interfaz de configuración de servicios

Además de los formularios se incluirá una serie de páginas de documentación y tutoriales asociados a cada servicio para facilitar que los clientes hagan un buen uso del sistema.

### **ASI 8.3 Especificación de Formatos Individuales de la Interfaz de Pantalla**

La interfaz de registro se compondrá de una serie de formularios en los que el usuario deberá introducir sus datos personales, así como un email y una contraseña

La interfaz de inicio de sesión se compondrá de dos formularios, uno para el correo y otro para la contraseña

En la interfaz del perfil de usuario se mostrarán los datos personales de dicho usuario y al lado de cada uno habrá un botón de editar dicho campo. Al pulsarlo el campo se convertirá en un formulario para poder modificar dicho dato.

La interfaz de método de pago se compondrá de una serie de botones-imagen de cada método de pago. Cuando el usuario pulse uno de los botones aparecerá un formulario para añadir la información requerida por dicho método de pago.

La interfaz de servicios muestra información y gráficas de los servicios activos que tiene el usuario. Además contiene un menú de configuración compuesto de formularios para editar los parámetros de dicho servicio.

La interfaz de selección y contratación de servicios muestra un catálogo con todos los servicios disponibles, también contiene un formulario de búsqueda para acelerar el proceso.

### **ASI 8.4 Especificación del Comportamiento Dinámico de la Interfaz**

En el formulario de registro se debe comprobar que el email introducido es válido. También se debe comprobar que la contraseña tiene al menos 8 caracteres, una mayúscula, un número y un signo de puntuación.

El formulario de inicio de sesión debe comprobar con la base de datos que el email introducido pertenece a un usuario existente y que la contraseña introducida se corresponde con dicho usuario.

Los formularios de edición del perfil del usuario deben realizar las mismas comprobaciones que el formulario de registro.

### **ASI 9.1 Verificación de los Modelos**

Una vez realizado el modelo de datos tenemos que asegurarnos de que los datos que vamos a guardar cumplen con la normativa local, en este caso con la GDPR.

### **ASI 9.2 Análisis de Consistencia entre Modelos**

Durante la elaboración del modelo conceptual de datos se han identificado las siguientes identidades:

Datacenter, Empleado, Servidor, Cliente. Sin embargo, durante el modelado de clases no se han diseñado clases para todas estas entidades, y de igual manera, no todas las clases diseñadas tienen una entidad asociada.

Tras arreglar esto la lista de clases quedaría así

Datacenter:

- Localización
- Empleados
- Servicios disponibles
- Lista de servidores

Empleado:

- Nombre
- Fecha de nacimiento
- ID
- Puesto
- Horario
- Sueldo
- Estado

Servidor:

- IP local
- Lista de recursos hardware
- Servicios disponibles
- Estado

Cliente:

- Nombre
- Servicios contratados
- Método de pago

RecursoHardware

- Nombre

Servicio

- Nombre
- Descripción
- Precio
- Cada tipo de servicio hereda de esta clase

### **ASI 9.3 Validación de los Modelos**

El requisito de facilidad de aprendizaje está principalmente relacionada con el modelado de la interfaz, aunque no solo la interfaz es relevante para cumplir este requisito. También es necesario una buena documentación así como tutoriales de uso, esta información extra estará separada en una plataforma dedicada al aprendizaje.

Respecto al los requisitos de rendimiento y seguridad, están parcialmente cubiertos por el modelo conceptual del hardware, pero seria interesante añadir también un modelo conceptual y lógico del software, de esta forma podríamos representar el las actualizaciones y agujeros de seguridad.

### **ASI 9.4 Elaboración de Especificación de Requisitos Software (ERS)**

Los requisitos del software son los siguientes:

- Facilidad de recuperación desde un estado de error
- Baja latencia
- Altamente distribuido
- Capacidad de actualizarse de forma transparente al usuario final

### **ASI 10.1 Definición del Alcance de las Pruebas**

Las pruebas para el área de gestión de sistemas se limitarán a cubrir el correcto funcionamiento, implementación, seguridad y desempeño del entorno tecnológico. Sin embargo, en otras áreas, el equipo de administración del sistema deberá implementar la fase de prueba de manera adecuada.

Los recursos necesarios para la fase de prueba deben estar listos antes del inicio de la fase de construcción del sistema.

### **ASI 10.2 Definición de Requisitos del Entorno de Pruebas**

Los requisitos para poder testear el sistema, tanto para la gestión como para el resto de áreas, son los siguientes:

- Un servidor para alojar aplicaciones que debe contener al menos:

Un sistema operativo equivalente al utilizado en producción.

- Monitorizar para calcular el rendimiento y poder estimar lo que se necesita en producción.

Copia parcial de los archivos de planificación del personal para la prueba de datos y la carga inicial.

- Pequeña LAN que simula la red de producción para pruebas de integración.

Prueba de punto de acceso inalámbrico para verificar las necesidades de cobertura.

- Lector de huellas dactilares sencillo para pruebas de autenticación.

- Lector de huellas dactilares de control de acceso con el correspondiente juego de tornillos para realizar pruebas de seguridad en instalaciones de apertura y cierre.

- Un gestor de bases de datos como el que ya utiliza la empresa.

- Réplicas de sistemas comerciales ya implementados con los que interactúa, en nuestro caso solo la base de datos de personal.

### **ASI 10.3 Definición de las Pruebas de Aceptación del Sistema**

Condiciones para aceptar las pruebas propuestas por el equipo directivo

Entre los sistemas se encuentran los siguientes:

- Un empleado puede llegar a una instalación utilizando el sistema de control en menos de un segundo.

- El empleado puede iniciar sesión o salir del edificio en menos de un segundo.

- Validar los datos de los empleados, horarios y tareas asociadas.

- Solo se puede acceder a las áreas restringidas mediante huella digital o llave maestra.

- Diariamente se realiza una copia correcta de los importantes datos del sistema.

- Todas las estaciones instaladas pueden acceder a la red y aplicaciones publicadas en el servidor.

El cumplimiento de todos ellos Garantizará una buena instalación del entorno tecnológico y una buena gestión del mismo.

El permite el sistema de control de acceso el acceso de manera apropiada según el permiso del empleado.

- El sistema de control de acceso ya las puertas abiertas cuando no hay luz.

La información de cualquier disco duro está duplicada en otro disco.

### **ASI 11.1 Presentación y Aprobación del Análisis del Sistema de Información**

En este apartado se realizarán especificaciones más detalladas que las implementadas hasta el momento para el sistema de información a implementar. Nuevamente, se definirán las actividades en las que participa el departamento de sistemas.

### **DSI 1.1 Definición de Niveles de Arquitectura**

Se entiende por nodo cada partición física o parte significativa del sistema de información, con características propias de ejecución o función, e incluso de diseño y construcción.

El particionamiento físico del sistema de información se especifica identificando los nodos y las comunicaciones entre los mismos, con cierta independencia de la infraestructura tecnológica que da soporte a cada nodo.

Los niveles están físicamente separados y se ejecutan en máquinas diferentes. Un nivel puede llamar a otro nivel directamente o usar mensajería asíncrona (cola de mensajes). Aunque cada capa se puede hospedar en su propio nivel, esto no es necesario. Se pueden hospedar varias capas en el mismo nivel. La separación física de los niveles mejora la escalabilidad y la resistencia, pero también agrega latencia a la comunicación de red adicional.

### **DSI 1.2 Identificación de Requisitos de Diseño y Construcción**

Primero, definiremos los requisitos asociados con la arquitectura tecnológica definida anteriormente porque influyen en gran medida en el diseño del sistema. Para ello, partimos de los requisitos de desempeño que debe proporcionar cada uno de sus elementos constitutivos. Servidores de datos y aplicaciones.

Los servidores de aplicaciones y datos deben estar disponibles al menos durante el horario comercial para que los empleados puedan utilizar las aplicaciones alojadas.

Los servidores de aplicaciones deben poder escalar automáticamente en función del número de solicitudes y la carga de trabajo.

Solo el grupo de Administradores puede acceder legítimamente y / o modificar los servidores de aplicaciones y datos.

Unidades del sistema. En el caso de nuestro proyecto, el sistema contiene tres módulos diferentes: el primer subsistema vinculado al control de horarios del personal, el segundo vinculado al control de acceso a las diferentes áreas del edificio, y finalmente, un tercero que facilita administración. tareas de los trabajadores. El requisito principal es que cada uno se almacene en un servidor con ranuras, como se mencionó anteriormente, para separar los tres subsistemas de modo que si el servidor falla, no afectará a todo el sistema.

### **DSI 1.3 Especificación de Excepciones**

se propone su catalogación como ayuda para el diseño del sistema de información y como guía en la especificación técnica de las pruebas, al permitir la generación de algunos casos de prueba de forma inmediata. Dicho catálogo se va completando a partir de las actividades correspondientes al diseño detallado de los subsistemas. Las excepciones en la aplicación web en directo se notifican mediante Application Insights. Puede correlacionar las solicitudes con error con excepciones y otros eventos en el cliente y en el servidor, de modo que pueda diagnosticar rápidamente las causas.

DSI 1.4 Especificación de Estándares y Normas de Diseño y construcción

DSI 1.5 Identificación de Subsistemas de Diseño

### **DSI 1.6 Especificación Entorno Tecnológico.**

En esta etapa se definen los recursos necesarios para dar soporte al nuevo sistema de información, incluyendo una planificación de la evolución de los mismos en un futuro. El objetivo consiste en adquirir componentes y servicios cuya duración sea máxima.

Comenzamos analizando las necesidades de almacenamiento del sistema en cuestión. Para implementar y configurar los nodos en los servidores de datos vamos a montar un cluster utilizando la herramienta MySQL Cluster. A través de ella podremos definir las direcciones de los servidores de datos que almacenarán la información y los servidores de aplicaciones que serán los que accedan a dichos datos.

Si bien se trata de una base de datos distribuida cabe destacar que en ella se almacenarán los datos respectivos a los tres subsistemas identificados. Como todos ellos demandan un gran número de peticiones de lectura y escritura, especialmente los de control de horarios y accesos.

### **DSI 1.7 Especificación de Requisitos de Operación y Seguridad**

En esta etapa, el objetivo es definir los requisitos relacionados con las medidas de seguridad, asociadas a los datos y al sistema, que se llevarán a cabo, así como los relacionados con las diferentes operaciones que el propio sistema puede realizar.

Controle el acceso al sistema y sus recursos.

- Se definirán cuatro grupos de usuarios según los recursos a los que tengan acceso y los procesos activados según su rol y los permisos asociados a ellos.

desarrolladores. Este grupo de usuarios tendrá acceso al administrador de la base de datos con permisos para agregar, modificar y eliminar

estructura del sistema. De la misma forma también tendrás el código

La fuente de los tres módulos, diseños, análisis, catálogo de requisitos,

Planificación y resultados de desarrollo y pruebas.

Integración de datos y confidencialidad.

- Cada usuario solo podrá acceder a la información autorizada según el rol que se le asigne en el sistema.

Se cifrarán las transmisiones de datos de todo tipo.

Haga una copia de seguridad y restaure los datos.

- Se realizará una copia de seguridad completa del sistema la primera vez que se inicie este proceso. El objetivo es obtener una copia de la información más relevante para el sistema inicial, como los archivos de administración, configuración y datos actuales.

Recuperación de desastres.

El resto de las copias de seguridad realizadas periódicamente solo registrarán los cambios en los datos.

### **DSI 2.1Diseño de Subsistemas de Soporte**

Es importante considerar la reutilización a la hora de diseñar los subsistemas de soporte. Para su diseño se deben reutilizar subsistemas que ya existan en la instalación y se consideren viables. Además el diseño del subsistema se debe realizar teniendo en cuenta su posible reutilización en futuros sistemas.

Las particularidades que hay que tener en cuenta en el diseño de subsistemas de soporte son las siguientes:

Generalmente, será necesaria una descomposición de los subsistemas de soporte en servicios, entendiendo como tales módulos o clases independientes y reutilizables.

Se recomienda realizar una descripción de la interfaz y del comportamiento de cada servicio, previa a su diseño de detalle, que permita completar el diseño de los subsistemas específicos.

La especificación y diseño de cada servicio, módulo o clase, se realiza con las técnicas habituales de especificación y diseño de módulos o clases, o incluso opcionalmente, si la simplicidad de los elementos lo aconseja, otros lenguajes de especificación, pseudocódigo o lenguaje natural.

### **DSI 2.2Identificación de Mecanismos Genéricos de Diseño**

Identificar y diseñar, en el caso de no existir en la instalación, esqueletos, patrones de diseño o guías de diseño. Estos mecanismos genéricos se definen a partir del estudio de comportamientos comunes relacionados, generalmente, con gestión de transacciones, persistencia de datos, control y recuperación de errores, utilización de recursos comunes, etc.

Los mecanismos genéricos de diseño son de aplicación tanto en la definición de la arquitectura del sistema como en el diseño de detalle de los subsistemas específicos y de soporte. Las funcionalidades integradas se organizan en seis áreas funcionales: Operaciones, Aplicaciones, Almacenamiento, Redes, Proceso e Identidades. Se proporcionan detalles adicionales sobre las características y funcionalidades disponibles en la plataforma Azure para estas seis áreas de forma resumida.

### **DSI 3.1 Identificación de Clases Asociadas a un Caso de Uso**

Define la funcionalidad básica para implementar modelos como puntos de conexión de servicio Web en Azure Machine Learning.

El constructor `WebService` se usa para recuperar una representación en la nube de un objeto `WebService` asociado al área de trabajo proporcionada. Devuelve una instancia de una clase secundaria que corresponde al tipo específico del objeto `WebService` recuperado. La clase `WebService` permite implementar modelos de aprendizaje automático desde un `Model Image` objeto.

Para obtener más información sobre cómo trabajar con `WebService`, vea implementación de modelos con Azure machine learning.

### **DSI 3.4 Revisión de Subsistemas de Diseño e Interfaces**

Para un caso de uso hay que definir, además de los subsistemas y actores que intervienen en el mismo, los mensajes que intercambian los objetos de un subsistema con otro. Estos mensajes sirven para verificar y detallar las interfaces de cada subsistema, teniendo en cuenta todos los casos de uso en los que interviene, y completar de esta manera la definición de subsistemas establecida en la tarea Identificación de Subsistemas de Diseño.

De entrada :

Modelo de Casos de Uso

Especificación de Casos de Uso

Diseño de la Realización de los Casos de Uso

De salida :

Diseño de la Realización de los Casos de Uso

Definición a Nivel de Subsistemas e Interfaz

### **DSI 4.1 Identificación de Clases Adicionales**

Una clase describe un conjunto de objetos con propiedades (atributos) similares y un comportamiento común. Los objetos son instancias de las clases.

No existe un procedimiento inmediato que permita localizar las clases del diagrama de clases. Éstas suelen corresponderse con sustantivos que hacen referencia al ámbito del sistema de información y que se encuentran en los documentos de las especificaciones de requisitos y los casos de uso.

Dentro de la estructura de una clase se definen los atributos y las operaciones o métodos:

Los atributos de una clase representan los datos asociados a los objetos instanciados por esa clase.

Las operaciones o métodos representan las funciones o procesos propios de los objetos de una clase, caracterizando a dichos objetos.

Establecen una clasificación de todos los objetos que pueden aparecer en un modelo. Los tipos son:

Objetos Entidad.

Objetos límite o interfaz.

Objetos de control.

## **DSI 4.2Diseño de Asociaciones y Agregaciones**

Las asociaciones entre las distintas clases suelen corresponderse con expresiones verbales registradas en las especificaciones de requisitos. Una asociación se caracteriza por:

Rol, o nombre de la asociación, que describe la semántica de la relación en el sentido indicado.

Multiplicidad, que describe la cardinalidad de la relación, cuántas

instancias de una clase están asociadas a una instancia de la otra clase. Los tipos de

multiplicidad son: Uno a uno, uno a muchos y muchos a muchos.

Autoservicio bajo demanda. Los usuarios pueden proveerse de cómputo en la nube sin requerir interacción humana o con el mismo proveedor (Microsoft).

Acceso ubicuo a la red. ...

Agrupación de recursos independientes de la posición.

Azure Data Catalog permite ver los recursos de datos relacionados con un recurso de datos seleccionado.

En esta ruta de aprendizaje se le proporciona una introducción a los conceptos básicos de la informática en la nube y la arquitectura de Azure

## **DSI 4.3Identificación de Atributos de las Clase**

Los atributos proporcionan una manera de asociar la información con el código de manera declarativa. También pueden proporcionar un elemento reutilizable que se puede aplicar a diversos destinos.

Considere el atributo [Obsolete]. Se puede aplicar a clases, structs, métodos, constructores y más. Declara que el elemento está obsoleto. Es decisión del compilador de C# buscar este atributo y realizar alguna acción como respuesta.

En este tutorial, se le introducirá a cómo agregar atributos al código, cómo crear y usar sus propios atributos y cómo usar algunos atributos que se integran en .NET Core.

## **DSI 4.4Identificación de Operaciones de las Clases**

El servicio Azure Storage para Gridwch, Gridwch.SagaParticipants.Storage.AzureStorage, proporciona operaciones de blobs y contenedores para las cuentas de Azure Storage configuradas para Gridwch. Crear blob, Eliminar contenedor, Copiar blob o Cambiar nivel de acceso son algunos ejemplos de operaciones de almacenamiento.

Gridwch requiere que sus mecanismos de almacenamiento funcionen tanto para blobs en bloques como para contenedores de Azure Storage. Con clases y operaciones del servicio de almacenamiento distintas para blobs y contenedores, no hay ambigüedad sobre si una operación de almacenamiento determinada se relaciona con un blob o con un contenedor. Este artículo se aplica a blobs y contenedores, excepto en los casos en los que se indique.

## **DSI 4.5Diseño de la Jerarquía**

Se aplica a: Configuration Manager (rama actual)

Antes de instalar el primer sitio de una nueva jerarquía de Configuration Manager, le recomendamos que comprenda lo siguiente:

Las topologías disponibles en Configuration Manager.

Los tipos de sitios disponibles y sus relaciones entre sí.

El ámbito de administración que proporciona cada tipo de sitio.

Las opciones de administración de contenidos que pueden reducir el número de sitios que necesita instalar.

## **DSI 4.7Especificación de Necesidades de Migración y Carga Inicial de Datos**

Con base en el modelo de datos anterior definido por el grupo de desarrollo del sistema, se determinan las siguientes necesidades de migración y la carga de datos inicial.



Como el sistema a implementar es nuevo, la única información previa presente es la del personal. Sin embargo, está prerregistrado en una base de datos.

Datos en el centro de datos de una organización. Todo lo que tiene que hacer es agregar los campos "Horas" y "Autorizaciones" a la tabla de empleados de la base de datos.

Por lo tanto, la única actividad necesaria es cargar los datos sin procesar.

Descarga de datos inicial:

Las tablas que se deben completar al cargar datos sin procesar son: Hoja de horas, Tarea, Transferencia y EmployeeTieneTarea.

Los datos a ingresar son triviales porque las tablas y tareas ya existen aunque no se ingresen al sistema y deben ser ingresadas por los propios empleados y gerentes a través de las aplicaciones provistas por el sistema. El sistema también guarda automáticamente las firmas.

Por lo tanto, solo queda ingresar los horarios de los trabajadores en el sistema. Para ello, no se requiere el uso de herramientas o software porque se deben copiar los horarios reales de los trabajadores.

El orden en el que se cargan las tablas será en horas, tareas, relación EmployeeHasTarea y finalmente transferencias una vez iniciado el sistema.

La forma de asegurar que las operaciones se lleven a cabo con éxito será verificar y validar los datos registrados por los empleados.

### **DSI 5.1Diseño de Módulos del Sistema**

Azure Automation usa una serie de módulos de PowerShell para habilitar cmdlets en runbooks y recursos de DSC en configuraciones de DSC. Los módulos compatibles incluyen:

Azure PowerShell Az.Automation

Azure PowerShell AzureRM.Automation

Otros módulos de PowerShell

Módulo Orchestrator.AssetManagement.Cmdlets interno

Módulos de Python 2

Módulos personalizados creados por el usuario

### **DSI 5.2Diseño de Comunicaciones entre Módulos**

Se revisan los elementos que forman cada interfaz, sus características (que deben ser consistentes con los atributos con los que están relacionadas), su disposición. Se revisa la navegación entre ventanas de la interfaz de usuario.

será necesario crear una solicitud de revisión independiente. Los costos habituales de soporte se aplicarán a las preguntas de soporte técnico adicionales y problemas que no califican para esta revisión específica.

### **DSI 6.1Diseño del Modelo Físico de Dato**

Los pasos básico para el diseño del modelo de datos físicos son los siguientes:

Convertir entidades en tablas.

Convertir relaciones en claves externas.

Convertir atributos en columnas.

Modificar el modelo de datos físicos en función de las restricciones / requisitos físicos

### **DSI 6.2Especificación de los Caminos de Acceso a los Datos**

Azure ofrece una selección de bases de datos relacionales, NoSQL y en memoria totalmente administradas, que abarcan motores patentados y de código abierto, para adaptarse a las necesidades de los desarrolladores de aplicaciones actuales.

El subsistema de optimización de consultas NoSQL es parte hoy en día de todo sistema de administración de bases de datos relacional . Dada una petición al servidor, resolver el problema de la elección de un “Camino de Acceso” (“Access Path” en inglés) es de gran importancia para dicho componente, ya que se trata de seleccionar una serie finita de pasos para retornar un conjunto de tuplas (filas) de una o más relaciones (tablas o vistas).

### **DSI 6.3 Optimización del Modelo Físico de Datos**

Una vez que haya identificado el problema de rendimiento que tiene con Azure SQL Database y la instancia administrada de Azure SQL, lo ayudará a:

Modifique su aplicación y aplique algunas de las mejores prácticas que pueden mejorar el rendimiento.

Modifique la base de datos cambiando índices y consultas para trabajar de manera más eficiente con los datos.

En instancias de SQL Server locales tradicionales, el proceso de planeamiento inicial de la capacidad con frecuencia está separado del proceso de ejecución de una aplicación en producción. Primero se adquieren las licencias de hardware y productos y luego se ajusta el rendimiento.

Para aumentar el rendimiento, puede realizar ajustes adicionales en las aplicaciones para que tengan estas características:

Aplicaciones que tienen un rendimiento lento debido a un comportamiento "comunicativo"

Bases de datos con una carga de trabajo intensiva que no se admiten en una sola máquina

Aplicaciones que tienen consultas poco óptimas

Aplicaciones que tienen un diseño de acceso a datos poco óptimo

### **DSI 6.4 Especificación de la Distribución de Datos**

Todos los datos de la aplicación serán almacenados en el mismo nodo, en este caso, el nodo servidor, donde se

encuentran tanto el código de la aplicación como los datos en la base de datos MySQL.

Los detalles siguientes no se pueden cambiar una vez creada una fuente de distribución de datos.

Identificador de la fuente de distribución de datos

Hora de creación

Dimensión

Tipo de origen

Granularidad

### **DSI 7.1 Verificación de las Especificaciones de Diseño**

Dadas las especificaciones del diseño y las necesidades que tendrá el sistema, la arquitectura x86 es la opción correcta para la implementación. La elección de sistema operativo Linux para todo el sistema de virtualización también es óptima. Para los procesadores y GPUs se recomienda realizar pruebas y tests de carga para decidir cuáles emplear.

En cuanto a los modelos de datos y especialmente la GUI, será recomendable realizar test de adaptación y test de calidad visual en la interfaz, pues una interfaz gráfica comprensible e intuitiva será clave para la buena aceptación del producto al mismo nivel que la calidad del producto.

### **DSI 7.3 Aceptación de la Arquitectura del Sistema**

El sistema es presentado al jefe de proyecto adjuntando el análisis de consistencia, donde se detalla en un nivel accesible la consistencia del proyecto. Haciendo especial incapié en el

análisis de requisitos y en cómo la estructuras y arquitecturas seleccionadas se ajustan a los mismos.

Con la implementación planeada se prevé que el usuario sea capaz de hacer uso del modelo de negocio IaaS en un comienzo para terminar siendo parte del entorno de SaaS donde en un futuro se ofertará gran variedad de productos.

### **DSI 8.1 Especificación del Entorno de Construcción**

En las especificaciones de entorno se tiene un entorno hardware donde se ha seleccionado la arquitectura x86, donde se realizarán test de carga sobre los dos tipos de procesadores disponibles, AMD e Intel, sin embargo se tomará la mayoría de AMD.

Se realizará una capa de virtualización y una capa de abstracción sobre el software de virtualización. Dada la necesidad de eficiencia y eficacia, ambas serán realizadas en el lenguaje C.

Se dispondrá de las mejores capacidades de red en el sistema, pues un requisito clave es que el usuario interactúe en “tiempo real” con su máquina virtual. Además se tendrá una disponibilidad de almacenamiento y procesamiento igual a 1.5 veces la ofertada, así, cada usuario tendrá la sensación de poder aumentar su procesamiento y almacenamiento en tiempo real y sin límites.

### **DSI 8.2 Definición de Componentes y Subsistemas de Construcción**

Se tendrá, como se ha mencionado anteriormente, una capa de virtualización, por encima de la misma, una capa software que interactúe con la virtualización y por encima de la misma una capa donde se creará la GUI con la que el usuario interactuará.

La capa intermedia se desarrollará manteniendo siempre test rígidos comprobando que el nuevo software y/o acciones no perjudicarán a los sistemas o máquinas que se encuentren en funcionamiento en el momento.

La capa en la cual se desarrollará la GUI seguirá el mismo funcionamiento, en general, todo procedimiento será test-driven. Así se evitarán posibles errores en la implementación o en la adición o borrado de máquinas virtuales.

### **DSI 8.3 Elaboración de Especificaciones de Construcción**

Dados los requisitos y como se ha mencionado anteriormente, la implementación será en C, las bases de datos se implementarán en SQL empleando índices para la mayor rapidez, aunque esto puede repercutir en un mayor almacenamiento, es lo indicado dadas las restricciones de tiempo con las que contará el sistema.

La interfaz se realizará con HTML, CSS y JavaScript, intentando desarrollar de la forma más rápida y eficiente evitando carga excesiva de módulos como podría tener node.js por ejemplo.

En general, toda la implementación se realizará en el nivel más cercano a la máquina que se encuentre.

### **DSI 8.4 Elaboración de Especificaciones del Modelo Físico de Datos**

Para el modelo físico de datos, como se menciona en la sección anterior se empleará SQL junto con índices sobre la base de datos con los cuales se ampliará la velocidad de búsqueda e inserción. Se desarrollará un optimizador de queries específico para el tiempo de queries presentadas y para este optimizador se realizará documentación para especificar las consultas, además de probar diferentes tipos de consultas para cada tarea para ver cual es mejor.

Para evitar problemas en el entendimiento con los trabajadores de Microsoft de otras áreas, se empleará la estructura de sistemas y carpetas con las que ellos trabajen en el momento, así como su mismo sistema de control de versiones.

### **DSI 9.1 Especificación del Entorno de Migración**

Para el entorno de migración, dado que se iniciará un sistema completo desde cero, lo importante será realizar de forma correcta el capacity planning. Saber cuanto, en capacidad de cómputo y almacenamiento, cuesta cada cliente para así adecuarse a ellos de la mejor forma. Dado que el cobro será por uso de máquina virtual, será necesario siempre tener recursos para todos los usuarios, y así calcular una tarifa de coste.

Se realizará un test donde empleados de otras secciones de microsoft tendrán acceso al sistema y podrán demandar recursos. Así se puede escalar o extrapolar este test para realizar previsiones del caso real de uso.

### **DSI 9.2 Diseño de Procedimientos de Migración y Carga Inicial**

El test inicial se hará como se indica en la sección anterior, las pruebas de seguridad se harán automáticamente pues se ha seguido una filosofía test-driven. En cuanto a copias de seguridad y protección de datos, estos procesos se realizarán en caliente y todos los servidores tendrán un backup equivalente a 3.5 veces la capacidad de almacenamiento del servidor. Es imperativo que no se de ni pérdida de datos ni errores en la asignación de usuarios y máquinas virtuales.

Los procesos de carga se realizarán priorizando la integridad de la información y que los datos relativos a los usuarios no tengan incongruencias.

### **DSI 9.3 Diseño Detallado de Componentes de Migración y Carga Inicial**

Como se ha mencionado en anteriores secciones el primer test de aproximación a la carga real se realizará con los empleados de otros sectores de la empresa. En este test se pondrá especial hincapié los tiempos de respuesta de los sistemas frente a los usuarios, a la gestión de copias de seguridad en caliente, a los sistemas de alerta en los servidores en cuanto a los componentes y a los componentes de la red.

A parte de los test ya realizados y los que se realizarán automáticamente, se crearán tests para cada subsistema y sus conexiones para evaluar los tiempos de respuesta, por ejemplo, se pueden realizar una serie de queries el doble de rápido de lo que lo haría un usuario normal para ver la respuesta del sistema. Todos los sistemas y subsistemas que pasen esta serie de test serán movidos a producción.

### **DSI 9.4 Revisión de la Planificación de la Migración**

El plan de migración se realizará con la filosofía de la sección 9.3, es decir, además de los test que se realizan durante la implementación se realizarán test de carga y test de parada del sistema en el caso de que la carga sea inusualmente alta. Para cada uno de los sistemas, incluyendo la GUI y la interacción con el usuario.

### **DSI 10.1Especificación del Entorno de Pruebas**

Como se ha mencionado anteriormente, el sistema dispondrá de un backup de al menos 2.5 veces el almacenamiento ofertado a clientes para realizar backups en caliente de los datos, esto se probará durante todas las fases del desarrollo. Al ser una arquitectura test driven, toda nueva lógica o procedimiento se testeará para comprobar que no existen conflictos. Esto se dividirá en test generales, test de subsistema y test de entorno. Como su nombre indica, cada uno de ellos irá desde el punto donde se introduce el nuevo código, hacia afuera. Si algunos de estos tests fallara se paraliza la salida a producción de la lógica o software. Contando así con los 3 entornos clásicos, desarrollo, test y producción.

### **DSI 10.2Especificación Técnica de Niveles de Prueba**

En este caso, se realizará un entorno donde se pruebe la funcionalidad de forma unitaria y aislada. Posteriormente pasará a un entorno donde se pruebe la funcionalidad dentro de su subsistema para pasar a un entorno de prueba de casos de uso donde se pruebe el subsistema en si con la nueva funcionalidad para comprobar que no afecta al rendimiento y eficiencia del subsistema por error.

Si todo esto ha ido bien, el código del subsistema nuevo se integrará en el global, para seguir con pruebas de sistema a nivel global, así, se asegura que el cambio software per se no ha afectado al subsistema ni al sistema global tanto en la lógica como rendimiento.

### **DSI 10.3Revisión de la Planificación de Pruebas**

Todos los test de software o lógicos serán llevados a cabo automáticamente, habiendo sido diseñados previamente por el manager de equipo correspondiente a cada subsistema y siendo responsabilidad exclusivamente del trabajador que ha implementado el código el solucionar el problema.

Por otro lado, los test de copias de seguridad y backups en caliente se realizarán varias veces por los miembros más experimentados de cada equipo, no necesariamente el jefe, y con el resto de miembros del equipo ofreciendo soporte.

### **DSI 11.1Especificación de Requisitos de Documentación de Usuario**

Dado que el usuario en última instancia tendrá contacto con el sistema GUI y la API de nuestros sistemas, esto es lo que debe estar documentado. Es importante que el nombre de las secciones no cambie para mantener la consistencia y coherencia, por ejemplo si una sección se llama “ampliar almacenamiento y recursos” no cambie su nombre entre versiones a “añadir recursos” pues el usuario no está presente en la implementación y no tiene por qué estar al tanto.

Dado esto, todas las funcionalidades del sistema a las que el usuario tenga acceso deben estar documentadas en la forma clásica de API, función, argumentos etc. Esta documentación se

organizará de igual manera que los sistemas y subsistemas en la aplicación, así si el usuario de la virtualización tiene dudas sobre cómo organizar datos deberá acudir al subsistema de almacenamiento y bases de datos.

### **DSI 11.2Especificación de Requisitos de Implantación**

Para la implantación será necesario ampliar la plantilla en el momento de salida a producción del producto, pues la demanda de trabajo será inevitablemente mayor, así como aumentar el personal de soporte esencial para la buena aceptación, es inevitable que existan errores en el futuro o que los usuarios necesiten un tiempo de adaptación.

El personal técnico recibirá formación sobre el entorno y los métodos de producción y ciclo de vida de un proceso. Por otro lado, el personal de soporte recibirá una formación básica del sistema, de sus subsistemas y errores comunes, así, cuando se presente una situación de error o necesidad de ayuda de un cliente, será más fácil para el personal de soporte proporcionar soporte o saber a quién acudir.

### **DSI 12.1Presentación y Aprobación del Diseño del Sistema de Información**

La directiva aprueba el proceso con connotaciones legales y de seguridad social por cambios de divisa.

### **CSI 1.1Implantación de la Base de Datos Física o Ficheros**

Se empleará SQL y como SGBD se empleará MySQL por compatibilidad y extensión del mismo, como se ha mencionado anteriormente en este documento, se emplearán tablas para representar a los usuarios, su información y sus sistemas asociados, por el lado propio de la empresa, es decir hardware y software, se emplearán tablas para registrar los sistemas en uso y también para controlar los espacios asignados a cada usuario.

La organización de la base de datos se hará por roles, así cada usuario, dependiendo de su posición o pago de recursos dispondrá de un rol. Por su parte, el equipo técnico también dispondrá de roles para gestionar el manejo de información y el acceso a la misma.

La estructura de ficheros será jerarquizada y cada jefe de subsistema podrá instaurar la que crea más adecuada para sus ciclos de vida de programación.

### **CSI 1.2Preparación del Entorno de Construcción**

Será esencial un equipo de gente especializada en virtualización y el lenguaje de programación C. Dadas las arquitecturas seleccionadas, será necesario que los líderes de los equipos de sistemas y virtualización colaboren como un equipo, sin descuidar sus tareas previas, para sacar el máximo rendimiento a la arquitectura x86.

Durante todo el proceso es importante la realización de test a medida que avance el proyecto para seguir la filosofía general del proyecto.

### **CSI 2.1Generación del Código de Componentes**

La generación de código se hará conforme al catálogo de normas siguiendo un sistema de revisión por parte de otros empleados, así, empleados podrán dedicar 30 minutos o 1h diarios para revisar código de sus compañeros con la intención de que el código sea lo más limpio, eficiente y escalable posible.

### **CSI 2.2 Generación del Código de los Procedimientos de Operación y Seguridad**

La generación de códigos de seguridad se hará, además de con los tests automatizados, con intención de detectar posibles fugas de datos o entradas para ataque, así como estrategias para su prevención.

### **CSI 3.1 Preparación del Entorno de Pruebas Unitarias**

Como se ha indicado en la revisión de planificación de pruebas, es conveniente que las pruebas a los dispositivos hardware se realicen varias veces. En cuanto al software, el desarrollo de test se realizará por cada equipo y entre equipos cuando los tests lo requieran.

Así, cada funcionalidad será testeada unitariamente, dentro de su subsistema y dentro del sistema global. De la forma más modular posible para evitar conflictos.

### **CSI 3.2 Realización y evaluación de las Pruebas Unitarias**

Las pruebas unitarias testearán si la funcionalidad es la deseada desde varios puntos de vista, si no se aprueban todos los tests diseñados para comprobar la robustez y resiliencia del sistema, la funcionalidad quedará parada durante un periodo máximo de 1 mes y si no es resuelto se dará el problema a un trabajador más cualificado.

### **CSI 4.1 Preparación del Entorno de las Pruebas de Integración**

Las pruebas de integración se realizarán siempre en el entorno de testeo diseñado y bajo un caso “sample” o de ejemplo que sea lo más real posible. Es decir, un caso donde se tengan datos como los reales pero sin que sean datos de ningún cliente o datos que revelen información sobre la empresa.

Además de estas pruebas, también se realizarán pruebas de aborto o pruebas de parado, es decir, tests donde se simule un caso de mal uso de datos o mal funcionamiento del sistema para comprobar la reacción tanto de los sistemas hardware como del software a estos problemas.

### **CSI 4.2 Realización de las Pruebas de Integración**

La realización de las pruebas será documentada, antes para su procedimiento, durante, para tener un timestamp del proceso y después para diseñar mejores pruebas en el futuro. Finalmente se creará un informe sobre las pruebas y sus resultados que será evaluado por los diferentes equipos, y finalmente por el jefe de proyecto.

### **CSI 4.3 Evaluación del Resultado de las Pruebas de Integración**

Si hubiera errores que no permitan el correcto funcionamiento del sistema en tiempo real, o como se espera ofrecer el servicio, la funcionalidad quedará parada para su revisión y mejora.

Sin embargo, si se presentan errores dentro de lo esperable la funcionalidad pasará a un entorno de prueba global mientras se es revisada.

### **CSI 5.1Preparación del Entorno de las Pruebas del Sistema**

Este tipo de pruebas será realizado con la misma metodología que las pruebas de integración, es decir, se dispondrá de unos datos de prueba a nivel de sistema donde se evaluará el correcto funcionamiento como esté esperado y si se dieran fallos quedaría relegado al subsistema o subsistemas correspondientes su correcta reparación.

### **CSI 5.2Realización de las Pruebas del Sistema**

Estas pruebas serán documentadas del mismo modo que las pruebas de integración y los informes irán dirigidos al jefe de proyecto y a los diferentes encargados de cada subsistema.

### **CSI 5.3Evaluación del Resultado de las Pruebas del Sistema**

Como en las pruebas de integración, si sus resultados son esperados se integrará en el global del sistema, sin embargo si hay fallos que interrumpan la experiencia de usuario o el funcionamiento del sistema, el o los subsistemas correspondientes serán reevaluados en otra iteración.

Si por otro lado, los fallos que se encuentren no interrumpen el funcionamiento esperado, esto se integrará en el sistema mientras los subsistemas tendrán la tarea de corregir esos fallos.

### **CSI 6.1Elaboración de los Manuales de Usuario**

Los manuales de usuario y documentación serán todos en formato web y públicos, tanto de los sistemas, API de subsistemas y guías de iniciación. Se seguirá el formato de documentación clásico de todas las API y funcionalidades.

La organización de esta documentación será jerárquica, así como lo es el sistema, así será más intuitiva de generar y mantene.

#### **CSI 7.1Definición del Esquema de Formación:**

Diseñar un esquema de capacitación para administradores de red que les permita potenciar los distintos servicios que se ofrecen en las bibliotecas para asegurar que el servicio sea altamente efectivo. objetivos específicos

Crear una herramienta que permita evaluar el nivel de conocimiento y alcance que poseen actualmente las herramientas de información de una red por parte del grupo de administradores que atienden directamente a los usuarios de la unidad de información.

Diagnosticar el conocimiento de las distintas fuentes de información de las unidades de información de la red.

Diseñar y modelar una propuesta de solución acorde a los resultados obtenidos para mejorar y mejorar el servicio brindado a los usuarios del servicio.

### **CSI 8.1Preparación del Entorno de Migración y Carga Inicial de Datos**



Se dispondrá el entorno en el que se construirán los componentes y procedimientos de migración y carga inicial de datos.

#### **Migración:**

Hay que preparar un entorno de migración para poder actuar en caso de tener que cambiar la localización física de los datos. Se tendrá que preparar scripts de automatización para desplegar la Base de Datos tanto en el CPD de la empresa como en Cloud. Se deberá de preparar la automatización del despliegue de la Base de Datos en servicios Cloud de varios proveedores, como pueden ser AWS, Azure y Google Cloud. Además también se deben tener preparados varios scripts para poder pasar toda la información almacenada a varios gestores de bases de datos diferentes, como puede ser MariaDB, que es el que se utilizará normalmente, MongoDB, PostgreSQL, o algún otro. Además de tener scripts de aprovisionamiento para aprovisionar al servidor de los gestores de Bases de Datos y las bibliotecas que necesiten. Además se deberá cargar toda la información que se ha almacenado hasta el momento. Para realizar todas estas tareas los desarrolladores tendrán que crear varios playbook de Ansible para poder obtener automáticamente todas las herramientas y bibliotecas necesarias. Estos playbook estarán almacenados en el repositorio privado de la empresa. De esta forma, cuando los trabajadores vayan a comenzar a trabajar sólo deberán ejecutar el playbook para tener sus máquinas listas para trabajar con las herramientas y bibliotecas necesarias.

**Carga Inicial:** En un principio la empresa solo tiene los datos de los empleados. Por lo que habría que crear las demás tablas (HORARIO, ESTADO, DEPARTAMENTO, TAREA, TAREA\_EMPLEADOS, CONTROL\_HORARIOS, CONTROL\_ACCESO) , que tendrían que ser rellenadas por los jefes de departamentos. En la carga inicial de los sistemas solo se incluirá la información de los empleados y la empresa. Los empleados rellenarán un formulario con sus datos. Se comprobará que los datos introducidos son correctos y se insertarán en el sistema.

#### **CSI 8.2 Generación del Código de los Componentes:**

En esta tarea se abordará la generación de los procedimientos de operación y administración del sistema de información, así como los procedimientos de seguridad y control de acceso, necesarios para ejecutar el sistema una vez que se haya implantado y esté en producción. Los desarrolladores deberán de seguir unos procedimientos de seguridad. Estos procedimientos estarán bien establecidos y documentados para que los desarrolladores puedan entenderlos perfectamente y además puedan volver a ser revisados tantas veces sea necesario. En estos procedimientos se incluirán aspectos como:

- Tener credenciales fuertes en su ordenador de trabajo
- Tener firewall y antivirus instalado y activado
- Tener configurado el antivirus para que haga una inspección del sistema de archivos al acabar la jornada laboral
- Utilizar únicamente sus propios ordenadores
- Crear copias de seguridad con AWS Backup. - Subir el código generado al repositorio al finalizar el día. Además de estos procedimientos, cada cierto tiempo se realizarán cursos sobre seguridad básicos en los que los trabajadores aprenderán a tener unas pautas de seguridad básicas. Estos cursos serán visualizados por los desarrolladores de manera online. También se realizarán pequeños cursos a medida que se vayan descubriendo nuevas estrategias para combatir riesgos nuevos.

#### **CSI 8.3 Realización y Evaluación de las Pruebas de Migración y Carga Inicial de Datos**

Esta actividad se centra en evaluar la disponibilidad de una carga de trabajo para migrar a la nube. Durante esta actividad, el equipo de adopción de la nube valida que todos los recursos y las dependencias asociadas sean compatibles con el modelo de implementación y el proveedor de nube elegidos. Durante el proceso, el equipo documenta los esfuerzos necesarios para corregir los problemas de compatibilidad. La mayor parte del contenido que analiza los principios de Cloud Adoption Framework no depende de la nube. Sin embargo, el proceso de evaluación de la disponibilidad debe ser muy específico para cada plataforma de nube específica. En la siguiente guía se presupone una intención de migrar a Azure. También se presupone el uso de Azure Migrate (también conocido como Azure Site Recovery) para las actividades de replicación. Para ver herramientas alternativas.

Actividades comunes de evaluación de la infraestructura:

- Requisitos de VMware: revise los requisitos de Azure Site Recovery para VMware.
- Requisitos de Hyper-V: revise los requisitos de Azure Site Recovery para Hyper-V.

Actividades comunes de evaluación de bases de datos:

- Documente los objetivos de punto de recuperación (RPO) y los objetivos de tiempo de recuperación (RTO) de la implementación de base de datos actual.
- Documente los requisitos de configuración de alta disponibilidad. Para entender los requisitos de SQL Server,
- Evalúe la compatibilidad de PaaS. En la guía de migración de datos de Azure se asignan bases de datos locales a soluciones de PaaS de Azure compatibles, como Azure Cosmos DB, Azure SQL Database, Azure Database for MySQL, Azure Database for PostgreSQL o Azure Database for MariaDB.
- Cuando la compatibilidad con PaaS es una opción sin necesidad de realizar ninguna corrección, consulte al equipo responsable de las actividades de arquitectura.
- Siempre que sea posible, documente las aplicaciones u otros recursos que realicen llamadas a cada base de datos.

### **CSI 9.1 Presentación y Aprobación del Sistema de Información**

En esta tarea se recopilan los productos del sistema de información y se presentan al Comité de Seguimiento para su aprobación. Con conectores de API, puede realizar la integración con sus propios flujos de trabajo de aprobaciones personalizados con el registro de autoservicio, para que pueda administrar qué cuentas de usuario invitado se crean en el inquilino.

En este ejemplo, el flujo de usuario de registro de autoservicio recopila datos de usuario durante el proceso de registro y los pasa al sistema de aprobación. Después, el sistema de aprobación puede:

Aprobar automáticamente el usuario y permitir que Azure AD cree la cuenta de usuario. Desencadenar una revisión manual. Si se aprueba la solicitud, el sistema de aprobación utiliza Microsoft Graph para aprovisionar la cuenta de usuario. El sistema de aprobación también puede notificar al usuario que se ha creado su cuenta

### **IAS 2.4 Seguimiento de la Formación a Usuarios Finales**

La lista de usuarios tanto participantes como finales del sistema junto con sus permisos es la siguiente.

- Administrador de servidores.
- Acceso a todas las instalaciones del complejo y a armarios de red. Permiso sobre el contenido y administración de las aplicaciones del sistema.
- Administrador de la base de datos.
- Acceso al contenido y administración de las bases de datos de las aplicaciones del sistema.
- Administrador de redes.
- Acceso a todas las instalaciones del complejo y a los armarios de red.
- Empleados.
- Sus privilegios pueden variar en función de su puesto y otros factores su matriz de privilegios se almacena en el sistema con los datos concretos. En ningún caso contarán con acceso a los armarios de red. El acceso a las aplicaciones vendrá únicamente dado para un uso a nivel de usuario mediante sus credenciales de acceso y las funcionalidades a las que puedan acceder dependerán de su puesto.

### **IAS 6.3 Evaluación del resultado de las Pruebas de Aceptación**

Se evalúan los resultados de las pruebas, analizando las incidencias recibidas y comprobando que se han llevado a cabo todos los casos de pruebas establecidos en el plan de pruebas. Dicha evaluación consiste en:

- Comparar los resultados obtenidos con los esperados.
- Identificar el origen de cada problema para poder remitirlo a quién proceda y determinar qué acciones o medidas correctoras es preciso llevar a cabo para resolverlo de forma satisfactoria.
- Indicar qué pruebas se debe volver a realizar, o si será necesario contemplar nuevos casos de prueba.

Una vez realizadas las medidas correctoras necesarias, y comprobado que su comportamiento es adecuado, se documenta el resultado global de la evaluación de las pruebas de aceptación que incluye la aprobación del sistema por parte del usuario final.

El objetivo de esta tarea es analizar los resultados obtenidos en las pruebas de aceptación y comprobar que son satisfactorios.

El propósito de este entregable es evaluar los resultados de la ejecución de las pruebas de aceptación y comprobar que se satisfacen las verificaciones especificadas en el plan de pruebas, de forma que se asegura que el sistema satisface las necesidades del cliente.

### **IAS 8.1 Identificación de los Servicios**

El servicio de identidad empresarial de Azure Active Directory (Azure AD) brinda inicio de sesión único y autenticación multifactor para ayudar a proteger a sus usuarios del 99,9 % de los ataques a la ciberseguridad. Se identifican los tipos de servicio requeridos por el sistema objeto de la implantación, en función de los sistemas de información que componen el sistema, sus requisitos y su localización geográfica. Entre los tipos de servicio se pueden distinguir los servicios al cliente (servicio de atención a usuario, etc.) y servicios de gestión de operaciones:

- Conecte a sus trabajadores
- Elija entre miles de aplicaciones SaaS
- Proteja y controle el acceso
- Interactúe con sus clientes y asociados
- Integre la identidad en sus aplicaciones

### **IAS 8.2 Descripción de las Propiedades de cada Servicio**

Para cada tipo de servicio identificado anteriormente se detallan sus propiedades funcionales, estableciendo las características que permiten especificar el funcionamiento del servicio (agentes que intervienen, acciones que se llevan a cabo, condiciones de activación, etc.).

- Conecte a sus trabajadores:

Tanto si los usuarios están en la oficina como si trabajan a distancia, proporcióneseles acceso sin problemas a todas las aplicaciones para que puedan seguir siendo productivos desde cualquier lugar. Automatice los flujos de trabajo para el aprovisionamiento y ciclo de vida de los usuarios. Ahorre tiempo y recursos con la administración de autoservicio.

- Elija entre miles de aplicaciones SaaS:

Simplify single sign-on. Azure AD supports thousands of pre-integrated software as a service (SaaS) applications.

- Proteja y controle el acceso:

Proteja las credenciales de los usuarios al aplicar una autenticación sólida y directivas de acceso condicional. Administre de forma eficaz sus identidades al garantizar que las personas correctas cuenten con el acceso adecuado a los recursos apropiados.

- Interactúe con sus clientes y asociados:

Proteja y administre a los clientes y asociados más allá de los límites de su organización, con una solución de identidad. Personalice el recorrido de los usuarios y simplifique la autenticación con una identidad social, etc.

- Integre la identidad en sus aplicaciones:

Acelere la adopción de sus aplicaciones en la empresa al admitir el inicio de sesión único y el aprovisionamiento de usuarios. Disminuya la fricción en el inicio de sesión y automatice la creación, la eliminación y el mantenimiento de las cuentas de usuario.

### **IAS 8.3 Determinación del Acuerdo de Nivel de Servicio :**

Una vez que el sistema se encuentra en su entorno de operación y ha sido aceptado se establecen formalmente los tipos de servicios a los que se debe dar respuesta, tanto por operación como por el usuario, mediante la especificación del acuerdo de servicio. En esta especificación se recoge el compromiso adquirido para cada tipo de servicio en términos de cumplimiento de los objetivos de nivel de servicio, considerando los recursos, plazos, coste, etc. Se determinan los mecanismos de regulación de los niveles de servicio. Mencionamos algunos de Acuerdo de Nivel de Servicio:

Azure HPC Cache: Azure HPC Cache no ofrece un acuerdo de nivel de servicio respaldado financieramente.

Azure Kubernetes Service (AKS): Por ser un servicio gratuito, AKS no ofrece un acuerdo de nivel de servicio respaldado financieramente. Nos esforzaremos por alcanzar al menos el 99,5 % de disponibilidad para el servidor API de Kubernetes. La disponibilidad de los nodos agente en su clúster está cubierta por el SLA de Máquinas Virtuales.

Service Fabric: Service Fabric es un servicio gratuito y, por lo tanto, no tiene un acuerdo de nivel de servicio con apoyo financiero. La disponibilidad del clúster de Service Fabric se basa en el SLA de las máquinas virtuales subyacentes y los recursos de almacenamiento utilizados.

Windows Virtual Desktop: Para los requisitos del servicio Windows Virtual Desktop, consulte los términos de la licencia descritos aquí. Microsoft no ofrece un acuerdo de nivel de servicio respaldado financieramente. Nos esforzaremos por alcanzar al menos el 99,9 % de disponibilidad para las URL del servicio Windows Virtual Desktop. La disponibilidad de máquinas virtuales de host de sesión en su suscripción está cubierta por el SLA de Máquinas Virtuales.

### **IAS 1.1 Definición del Plan de Implantación**

Se realizará un plan de implantación en el que se seguirá un determinado orden. En primer lugar, se formará a los miembros de los diferentes equipos para que entiendan perfectamente su labor en la tarea de la implantación. Ocasionalmente se deberán de ayudar los miembros de diferentes equipos para informar a los demás sobre qué aspectos de su entorno deben de tener en cuenta. Una vez que se han planteado correctamente los cambios necesarios, se prepara toda la infraestructura para poder realizarlos. Después se instalan los componentes y se realiza una carga inicial de datos en la Base de Datos de prueba, puesto que los datos en producción se generarán una vez el sistema esté en marcha. Finalmente se formará el plan de mantenimiento.

**IAS 1.2 Especificación del Equipo de Implantación:** El equipo de implantación estará formado por:

- Encargados del CPD: Serán el administrador de servidores y el administrador de la Base de Datos, que son los únicos que tienen permiso para poder acceder al CPD.
- Desarrolladores del software: En la etapa de implantación solamente deberán de informar a los encargados del CPD sobre la instalación de los componentes para que los instalen, informar al responsable de mantenimiento de los procesos que se realizan en cada subsistema y de especificar los recursos de red que necesitan para trabajar al encargado de red.

- Desarrolladores de pruebas: Se encargarán de realizar las pruebas antes de pasar a producción.
- Encargados de red: Tendrá que ver que se cumplen todos los requisitos necesarios para satisfacer las necesidades de todos los equipos.
- Encargados de mantenimiento: Deberán de conocer el sistema antes del paso a producción para tener conocimiento de su funcionamiento antes de llegar a la tarea de mantenimiento.

## **IAS 2.2 Formación del Equipo de Implantación /formación neceser para la implementación**

El administrador de servidores y el administrador de la base de datos serán formados para realizar los nuevos cambios en el CPD, puesto que son los únicos que tienen acceso a él. Se explicará detalladamente los nuevos componentes necesarios. Se instalará un nuevo armario en el CPD para las aplicaciones de control de acceso, administración de tareas y fichaje. También se instalará otro armario para almacenar los datos de las aplicaciones. Deberán estimar correctamente los recursos aproximados que consumirán las aplicaciones una vez se encuentren en producción para seleccionar los componentes que mejor se adapten a las necesidades de procesamiento y memoria de forma que las aplicaciones siempre estén disponibles y sean fácilmente escalables.

El administrador de servidores deberá estudiar la documentación del software de cada subsistema creada por los desarrolladores para poder instalar las aplicaciones en el CPD. Conocerán las herramientas necesarias de cada aplicación. El administrador de la Base de Datos deberá conocer las herramientas necesarias para el despliegue de la Base de Datos y de la Base de Datos que se utilizará para las pruebas. Ambos administradores, además de estudiar toda la documentación creada por los desarrolladores, tendrán una semana en la que se reunirán junto a los desarrolladores para resolver posibles dudas y facilitar que los administradores comprendan todo a la perfección.

El administrador de redes deberá conocer el número de trabajadores que trabajará en cada subsistema y los lugares en los que trabajaran de forma que pueda estudiar los componentes hardware más adecuados para crear una red estable y que cumpla las necesidades de red de cada área, además de hacer la red escalable para poder ampliar la red sin problema en caso de que fuese necesario algún día.

El responsable de mantenimiento deberá tener un conocimiento general de las tareas que realizan cada uno de los subsistemas. Estudiará los procesos generales que realiza cada aplicación de forma que si algún día ocurriese cualquier incidente, sepa a que equipo debe dirigirse para comentar los problemas que han ocurrido para que se encarguen de solucionarlo. Tendrá unas semanas para conocer los diferentes subsistemas y reunirse con los demás empleados para que le ayuden con el aprendizaje. La finalidad del responsable de mantenimiento en esta etapa de implantación es que comprenda perfectamente el sistema antes de que comience su labor de mantenimiento.

## **IAS 3.1 Preparación de la Instalación:**

Se deberán obtener las terminales tipo tablets con sistema operativo android que utilizarán los empleados a la hora de gestionar las tareas. También se deberán obtener los lectores de huella dactilar que se incorporarán en las tablets y los que se utilizarán para el control de acceso.

También será necesario 3 nuevos blades en el CPD para las aplicaciones y otro para la Base de Datos. Se deberá obtener un router para la planta en la que se encuentran la oficina en la que estarán los desarrolladores.

Se creará un documento en el que se vayan apuntando las incidencias que surjan durante la preparación, registrando la gravedad del problema, la causa y la solución que se debe realizar.

Con este documento de incidencias se tratará de no olvidar acabar con todos los detalles durante la preparación, de forma que se minimice los problemas posteriormente.

### **IAS 3.2 Realización de la Instalación:**

Se obtendrán todas las herramientas necesarias para la instalación del control de acceso, el registro de horarios, la gestión de tareas y la Base de Datos. Se instalarán todos los sensores de huella dactilar en todas las áreas restringidas y en la entrada para el registro de horarios. Se crearán las nuevas tablas en la Base de Datos y los grupos de usuarios definidos anteriormente (4.3 Requisitos de operación y seguridad), en los que se establecerán los permisos que tendrán en función de los recursos a los que puedan acceder. Estos grupos serán los de 'Desarrolladores', 'Administrador', 'Gestores', 'Trabajadores'. Se debe configurar la automatización de la generación de copias de seguridad con el servicio AWS Backup, de modo que se realice y almacene una copia de cada subsistema los fines de semana, guardando tanto los datos actuales, como los ficheros de configuración y administración. También se instalará y configurará el sistema de discos en RAID en el CPD. Se instalarán las cámaras de vigilancia en las zonas establecidas.

### **IAS 4.1 Migración y Carga Inicial de Datos:**

Descubra en qué medida está preparado para iniciar la migración con la herramienta SMART, La herramienta de evaluación y preparación para la migración estratégica (SMART) le ayuda a averiguar todo lo que debe hacer para prepararse para la migración a Microsoft Azure, desde el planeamiento empresarial y el aprendizaje, hasta la seguridad y la gobernanza.

Prepárese hoy para los desafíos del mañana, La migración a la nube no es solo para mejorar la escalabilidad y la seguridad. También es para mejorar la resistencia financiera al cambiar de un modelo tradicional basado en el gasto inicial a un modelo más flexible de pago por uso.

En función del modelo de datos anterior especificado por el grupo de desarrollo del sistema se definen las siguientes necesidades de migración y carga inicial de datos. Ya que el sistema a implementar es nuevo, la única información previa existente es la de los empleados. No obstante, esta se encuentra almacenada de antemano en una base de datos en el centro de procesamiento de datos de la organización. La única necesidad es la de añadir los campos 'Horario' y 'Permisos' a la tabla de empleados de la base de datos. Por lo tanto, la única actividad necesaria es la de la carga inicial de datos. Carga inicial de datos: Las tablas a rellenar en la carga inicial de datos son: Horario, Tarea, Fichaje y EmpleadoTieneTarea. Los datos a introducir son triviales ya que los horarios y las tareas son ya existentes, aunque no estén introducidos en el sistema y deberán ser introducidos por los propios empleados y jefes de departamentos mediante las aplicaciones que ofrece el sistema. Los fichajes también se registran de manera automática por el sistema. Por lo tanto, queda solo introducir los horarios de los trabajadores en el sistema. Para ello no se requiere el uso de ninguna herramienta o software ya que deben ser copiados los horarios reales de los trabajadores. Se hará mediante el uso de un script que recoja los datos de los documentos Excel donde estén reflejados los horarios y los introduzcan en la base de datos. El orden de carga de las tablas será Horarios, Tareas, la relación EmpleadoTieneTarea y por último los fichajes una vez el sistema comience

su funcionamiento. El método de confirmación de que los procesos han sido realizados de manera exitosa será mediante contraste por medio de los empleados de los datos registrados y verificando que son correctos.

#### **IAS 5.1Preparación de las Pruebas de Implantación:**

La preparación de las pruebas de implantación comienza con la generación del entorno de pruebas. Para ello se deben ejecutar los playbook de Ansible generados por los desarrolladores de las pruebas unitarias, de integración y del sistema. De esta forma se instalarán automáticamente todas las herramientas necesarias para realizar las pruebas. También se asegurarán de que los datos están correctamente cargados en la Base de Datos de prueba antes de realizar las pruebas.

#### **IAS 5.2Realización de las Pruebas de Implantación:**

Una vez realizado el provisionamiento, se ejecutarán todas las pruebas unitarias, de integración y del sistema creadas. Se recopilarán todos los resultados para obtener el comportamiento del sistema frente a estas pruebas.

#### **IAS 5.3Evaluación del resultado de las Pruebas de Implantación:**

Se estudiarán los resultados que se han obtenido en la fase de realización de las pruebas de implantación, conociendo si todas las pruebas se han realizado correctamente. En caso de que se hayan obtenido errores a la hora de realizar los tests se deberán de anotar en un lugar accesible para todo el personal para que tengan conocimiento del error que existe en el momento. Estos errores deberán de estar bien comentados, teniendo la información necesaria sobre cómo de grave es, los subsistemas que se ven afectados, el tipo de tareas que no se pueden realizar por consecuencia de estos errores, y en caso de que exista conocimiento sobre ello almacenar también la información asociada a la causa que lo provoca y la solución que se debe dar. Estos errores deberán de tener una clasificación de gravedad del error para que los desarrolladores tengan un orden de prioridad a la hora de solucionarlos.

#### **IAS 6.1Preparación de las Pruebas de Aceptación:**

Pruebas de aceptación.

- Hardware. Se emplea el mismo que en las pruebas anteriores.
- Software. Se desarrollarán las pruebas de aceptación necesarias para verificar que el sistema cumple con los requisitos especificados y que todos los casos de uso definidos se encuentran incluidos en él. Para ello, de nuevo, se adquirirán las licencias de las librerías necesarias para implementar los tests.
- Comunicaciones. Se emplea el mismo esquema que en las pruebas anteriores.
- Planificación de las capacidades necesarias. Son las mismas que las especificadas anteriormente.
- Planificación de la información necesaria. Serán necesarios los datos de prueba, los casos de uso y los requisitos del sistema para comprobar que este los satisface.
- Requisitos de operación, seguridad y procedimientos de recuperación. Se aplican los mismos requisitos que en las pruebas anteriores

#### **IAS 6.2Realización de las Pruebas de Aceptación:**



Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario. Estas pruebas van dirigidas a comprobar que el sistema cumple los requisitos de funcionamiento esperado, recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de información, y conseguir así la aceptación final del sistema por parte del usuario. El responsable de usuarios debe revisar los criterios de aceptación que se especificaron previamente en el plan de pruebas del sistema y, posteriormente, dirigir las pruebas de aceptación final. El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

#### **IAS 7.1 Establecimiento de la Infraestructura para el Mantenimiento:**

El responsable de mantenimiento deberá comprobar si el entorno está preparado adecuadamente para la tarea de mantenimiento. Deberá conocer si existen actualmente herramientas en la organización para la gestión de mantenimiento y comprobar si son adecuadas. Si no lo son deberá de investigar cuáles son las que se deberían adquirir. Tienen que existir herramientas para: - Registrar peticiones de mantenimiento, evaluarlas, controlarlas, realizar cambios y asegurar que se implementan correctamente. - Comunicación con los demás empleados. - Herramientas de monitorización de máquinas. - Herramientas de alerta al detectar cualquier peligro como por ejemplo, detección de temperatura demasiado elevada en un componente del CPD.

#### **IAS 7.2 Formalización del Plan de Mantenimiento:**

El responsable de mantenimiento formará parte del Equipo de Implantación de forma que conozca el sistema antes que haya pasado a producción. Deberá conocer las necesidades de los usuarios que usan el sistema. Cada equipo mostrará su parte al responsable de mantenimiento para facilitar que entienda cada subsistema desde un mejor punto de vista.

El responsable deberá comentar a cada equipo si la información que le han proporcionado es suficiente para que él pueda valorar el estado del sistema para su futuro mantenimiento. De forma que sepa perfectamente si los productos están completos, actualizados y son consistentes y precisos.

Para el subsistema de registro de horarios, se realizará un mantenimiento correctivo, en el que se corregirán los errores conforme vayan apareciendo, como puede ser la necesidad de sustituir un lector de huellas que ha dejado de funcionar.

Para el subsistema de control de acceso, se realizará tanto un mantenimiento correctivo como preventivo. Mantenimiento correctivo porque si un empleado no tiene permiso para realizar una tarea que verdaderamente debería poder realizar se tiene que corregir una vez aparezca la situación. Mientras que el mantenimiento preventivo se encarga de comprobar cada cierto tiempo que los permisos que están establecidos para cada grupo de empleados es el correcto para que puedan realizar sus tareas, especialmente por seguridad.

Para el subsistema de gestión de tareas, se realizará un mantenimiento correctivo, de forma que se solucionen los problemas una vez los empleados o sus superiores detecten algún error en él. En cuanto al CPD se debe realizar todos los mantenimientos posibles:

- Correctivo: Corregir cualquier error que se produzca en el momento en el que aparece, como puede ser que haya que sustituir un componente, se encuentre alguna vulnerabilidad, se necesite aumentar el almacenamiento, etc.
- Preventivo: Se deben realizar pruebas regulares para comprobar el estado general, realizando pruebas de vulnerabilidades, comprobando el estado de los componentes, etc.
- Predictivo: Se debe monitorizar constantemente el CPD, obteniendo todos los valores posibles sobre su funcionamiento, temperatura, estado, etc. De esta forma se pretende evitar que suceda alguna consecuencia o estar totalmente preparados cuando vaya a suceder.

#### **IAS 9.1. Convocatoria de la Presentación del Sistema:**

En este paso se realiza una convocatoria para la presentación del sistema al Comité de Dirección y se espera la confirmación por parte del Comité de Dirección. Se debe preparar una presentación en la que se vea bien reflejado el estado del sistema. Entre los apartados que se tomarán habrá un informe detallado con la evaluación obtenida de las pruebas realizadas y el plan de mantenimiento, mostrando todos los procesos que se realizaron y que se han probado correctamente.

#### **IAS 9.2 Aprobación del Sistema:**

Se presenta el sistema al Comité de Dirección según el plan previsto y se aprueba formalmente el sistema.

#### **IAS 10.1 Preparación del Entorno de Producción:**

En este punto se comprobará que la instalación del sistema en producción es correcta. Se debe haber comprobado ya que los sistemas de control de acceso, gestión de tareas y registro de horarios funciona a la perfección en el entorno de pruebas, por lo que solo será necesario asegurarse de que la Base de Datos de producción está preparada para funcionar.

#### **IAS 10.2 Activación del Sistema en Producción:**

Se levantará el sistema utilizando la Base de Datos del entorno de producción, la cual en un principio solo contará con los datos de los empleados y se comenzarán a registrar la información que generará el uso de los subsistemas nuevos, como las tareas a realizar y a asociarlas a los empleados encargados de la tarea y los permisos de acceso que tienen. Una vez que el sistema en producción está activado, comenzaría el proceso de Mantenimiento.