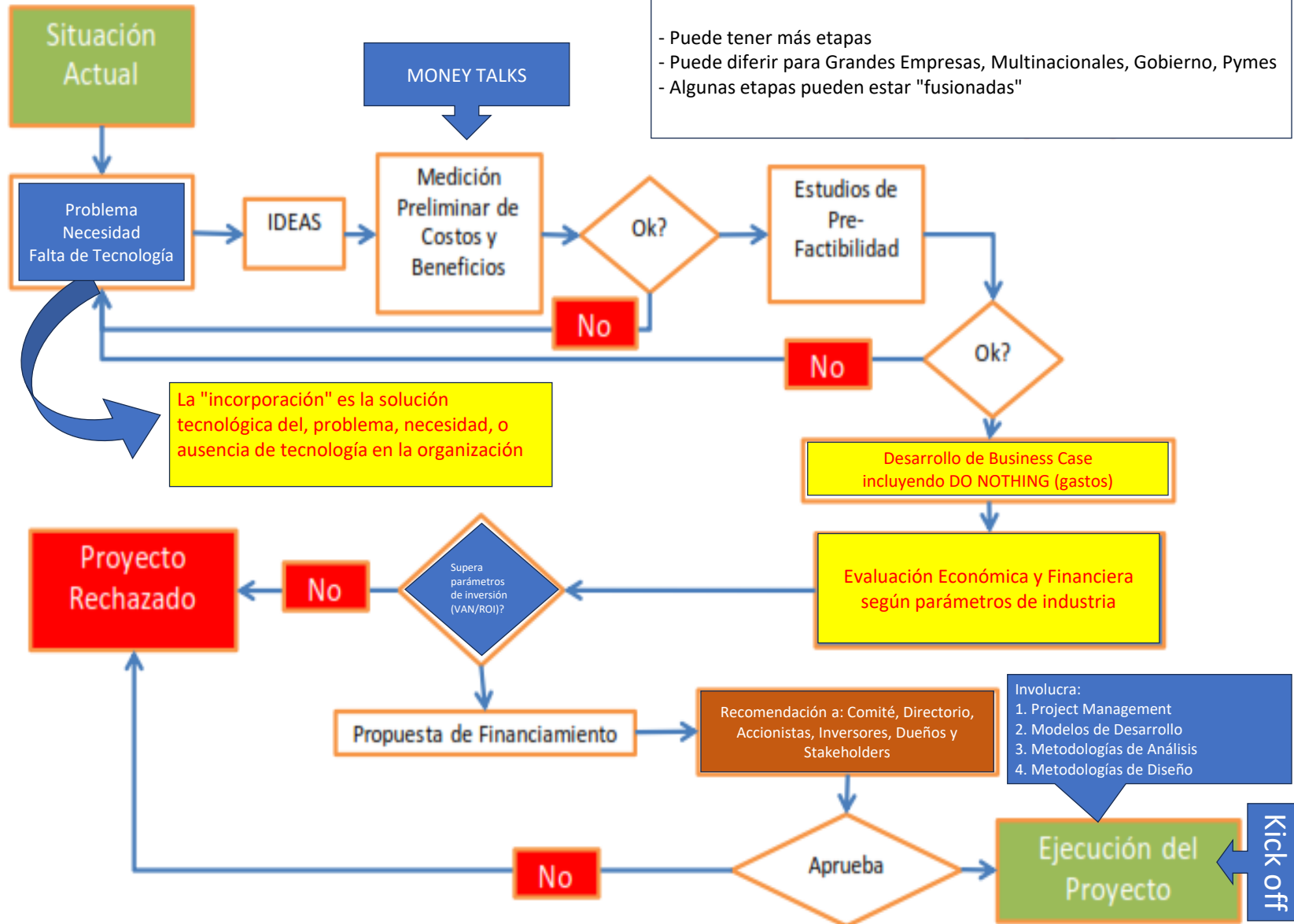
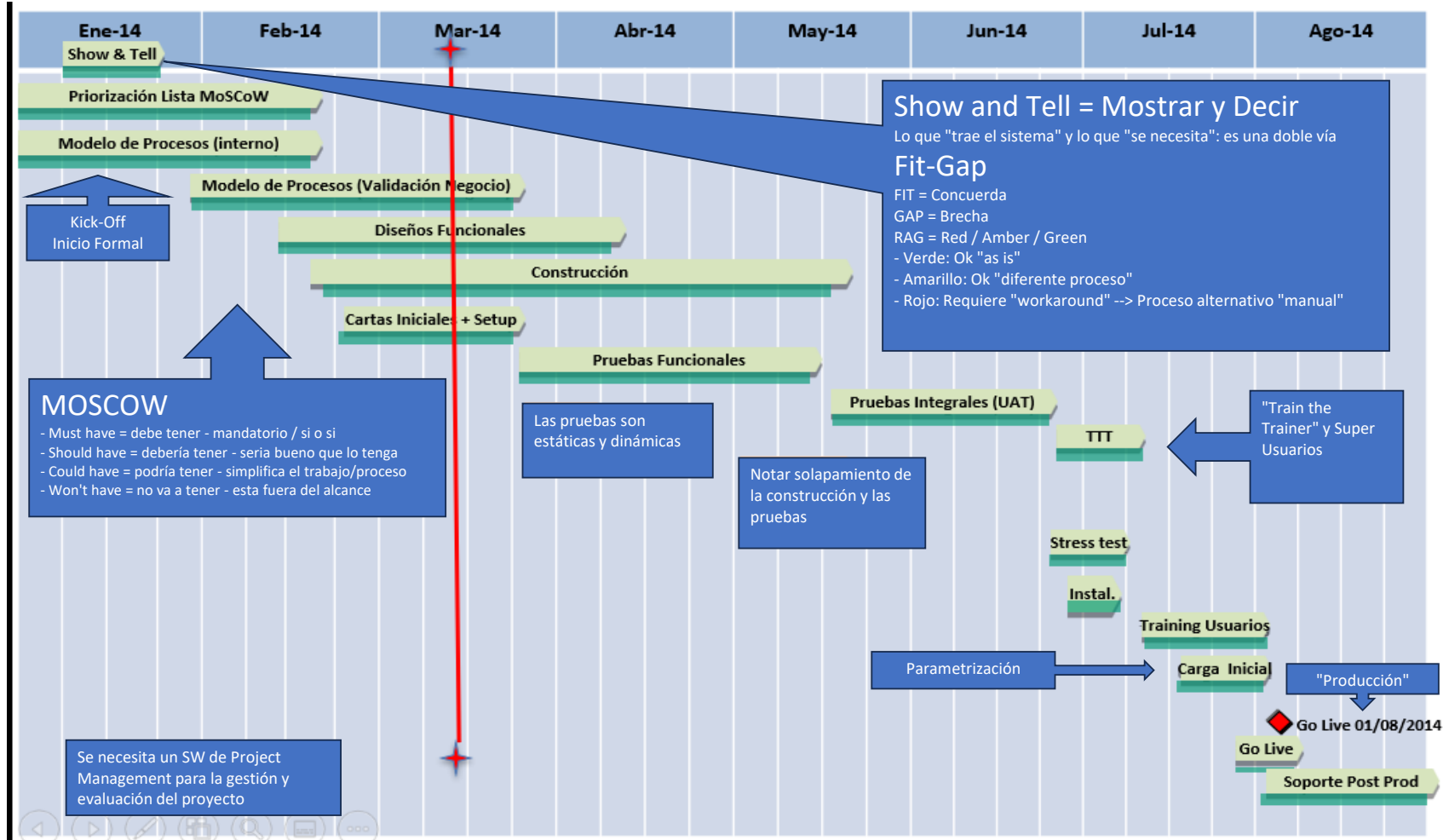


ESQUEMA DE EVALUACIÓN DE PROYECTOS

- Puede tener más etapas
- Puede diferir para Grandes Empresas, Multinacionales, Gobierno, Pymes
- Algunas etapas pueden estar "fusionadas"





INCORPORAR SISTEMAS DE INFORMACIÓN - Como - Cuando - Alternativas

100% a medida (custom)
100% estándar (enlatado)

¿Variables?

Insourcing
Outsourcing

HSBCPPSC
HSBCPPSC

Estandar + Custom

Webservices + API / REST

Hibridez en todos los componentes

On Cloud / On Premises

"DESPLIEGUE"

Costos en pesos / dolares

CONCEPTOS VARIOS

MOSCOW LIST

(PRIORIZAR)

must have
should have
could have
won't have



debe tener - mandatorio / si o si
deberia tener - seria bueno que lo tenga
podria tener - estaria bueno (porque simplifica...)
no va a tener - esta fuera del alcance

SHOW AND TELL

MOSTRAR
Y
DECIR



LO QUE TIENE EL SISTEMA
LO QUE NECESITAN DEL SISTEMA

TTT = TRAIN THE TRAINER



CAPACITAR AL CAPACITADOR

Análisis FIT/GAP

FIT = Concuerda

GAP = Brecha

Verde: Ok "as is"

Amarillo: Ok "diferente proceso"

Rojo: Requiere "workaround" --> Proceso alternativo "manual"

RAG = Red / Amber / Green



GAP

GAP

GAP

FIT

CORE / CUSTOM

Lo que cada organización requiere a medida

Lo específico de cada negocio

Lo legal de cada Pais

Core = Estándar

WORKFLOWS

Publico
Internas

Front Office
Back Office

Front End
Back End



Procesos de Punta a Punta

STP: Straight-through processing
Ver presentación en Econ 2024

¿Cómo es el proceso total?

On-line

Batch

Sincrónico - Asincrónico

INTEGRACION DE SISTEMAS

CONSIDERAR

CAPTURA
PROCESO
ALMACENAMIENTO
DISTRIBUCION



Centralizado
Descentralizado
Distribuida

Como puedo "integrar" en forma NO nativa

Traslado fisico de los datos
Electronic Data Interchange - EDI
Creacion de interfaces
Vistas de una Bases de Datos hacia la otra
Middleware
Webservices



Con o Sin destrucción de datos en origen

Software en el "middle" = medio

Temas a considerar en los Proyectos de Sistemas de Información

- | | |
|---------------------------------|--|
| 1) <u>Project Management</u> | Periodos de Freeze de sistemas |
| 2) <u>Modelos de desarrollo</u> | Estacionalidad de la operación del negocio |
| 3) <u>Análisis</u> | |
| 4) <u>Diseño</u> | |

Todo Proyecto de sistemas cuenta con 7 fases

(Otros autores y profesionales las agrupan de forma diferente)

1 - Estudios de Factibilidad

Factibilidad: Económica, Técnica, Operativa, Legal, Política, Social = Ídem análisis Pestel (Políticos, Económicos, Sociales, Tecnológicos, Ecológicos, Legales)

Análisis de HSBCPPSC

Generación del Business Case

Aprobación del Business Case

Kick-off - Puntapié inicial que realiza el inicio formal del proyecto

2 - Análisis y Diseño

Metodologías de análisis - Estructuradas / Orientadas a Objeto

YOURDON

Metodologías de diseño

(Si existe diseño...)

Etapas / Tradicional

Cascada

Espiral

Incrementales

Agiles

3 - Construcción y selección de tecnologías

Entornos de Construcción

Análisis de tecnologías requeridas

Testeos "alfa" y "beta"?

Ambientes

Desarrollo

Testing

QA - Quality Assurance

Training

Pre-Produccion

Producción

Simulación /

Post- Produccion

Programadores – WIP (Work in process)

Usuarios + Programadores

Aseguramiento de Calidad / Equipo QA o externos

Capacitación

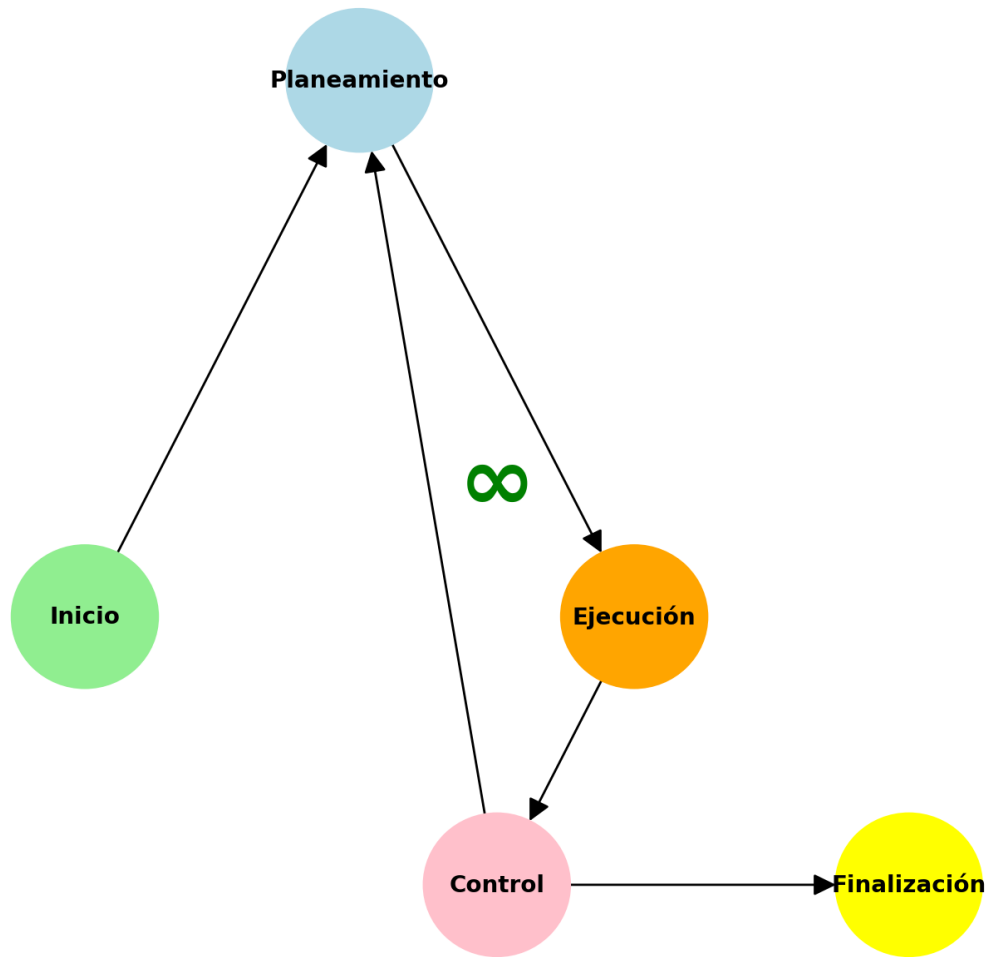
Simulacion con datos "CASI REALES" --> Disociados por "criticidad"

¡DATOS REALES! ----> IMPORTANTE

¡DATOS REALES!

SOW = STATEMENT OF WORK - Detalle de trabajo

Que / Quien / Cuando / Como / Para Qué - Por Qué



4 - Pruebas modulares, de sistemas y Quality Assurance

Pruebas Modulares	Estrategias	
Pruebas de Sistemas		Smoke Test
Pruebas estáticas		--> De área
Pruebas dinámicas	Carga Pico / Stress	--> De Circuito
Quality assurance como proceso		

Unidades/Unitarias:

Testear cada unidad o modulo por separado.

Pruebas de Sistema:

¿Funciona todo como se planeo?

Pruebas de Aceptación:

Certificación final – User Acceptance Test para pasar los desarrollos entre los ambientes

Plan de Tests:

Preparaciones para las pruebas a realizarse.

HSBCPPSC

\$ 1 = Licenciamiento y "fierros"

1 = 1,50

\$ 1,50 = Servicios adicionales (capacitación y el soporte)

Capacitación a usuarios

Estrategias de Conversión de Bases de Datos y formatos

Estrategias de Migración de datos maestros

Estrategias de "Volcado" de transacciones

Concepto	¿Qué implica?	¿Transforma a los datos?	¿Requiere sistema destino?	¿Se usa para...?
Conversión	Cambio de formato o tipo de datos	Sí	No necesariamente	Compatibilizar estructuras
Migración	Movimiento de datos entre sistemas	Sí (suele incluir conversión)	Sí	Sustitución de sistemas
Volcado	Extracción y almacenamiento en bruto	No	No	Respaldo, auditoría, análisis previo

1. Conversión de Datos

Definición: La **conversión** de datos es el proceso de **transformar datos de un formato a otro** para que puedan ser utilizados en un nuevo sistema o dentro del mismo sistema actualizado.

Ejemplos:

Cambiar fechas de formato DD/MM/AAAA a YYYY-MM-DD.

Transformar una columna de texto en valores numéricos codificados.

Pasar archivos .csv a una base de datos relacional como Oracle o SQL Server.

Objetivo: Asegurar que los datos sean **compatibles con el sistema destino** en cuanto a estructura, tipo y formato.

2. Migración de Datos

Definición: La **migración** implica **mover datos de un sistema a otro**, generalmente cuando se cambia de plataforma tecnológica, de versión o de proveedor de software.

Ejemplos:

Migrar datos de SAP a Oracle.

Trasladar un sistema local a una plataforma en la nube (on-premise a cloud).

Cambiar de un sistema legacy (antiguo) a uno moderno.

Incluye:

Extracción de datos.

Transformación (conversiones incluidas).

Carga en el nuevo sistema (**ETL** – Extract, Transform, Load).

Validación y pruebas de integridad.

Objetivo: Cambiar de entorno tecnológico asegurando **mínima pérdida de datos y compatibilidad funcional**.

3. Volcado de Datos (Dump)

Definición: El **volcado de datos** es el proceso de **extraer y guardar datos en bruto** desde una base de datos o sistema, generalmente en un archivo externo, sin necesariamente transformarlos.

Ejemplos:

Exportar toda una base de datos a un archivo .dmp en Oracle.

Crear un backup plano de una base de datos en formato .sql o .csv.

Dump de logs del sistema o de tablas para auditorías.

Usos comunes:

Copias de respaldo.

Transferencias masivas.

Auditorías o revisiones técnicas.

Cargas posteriores (como parte de una migración, pero no siempre).

Objetivo: Extraer información completa para **almacenamiento, análisis o traslado posterior**.

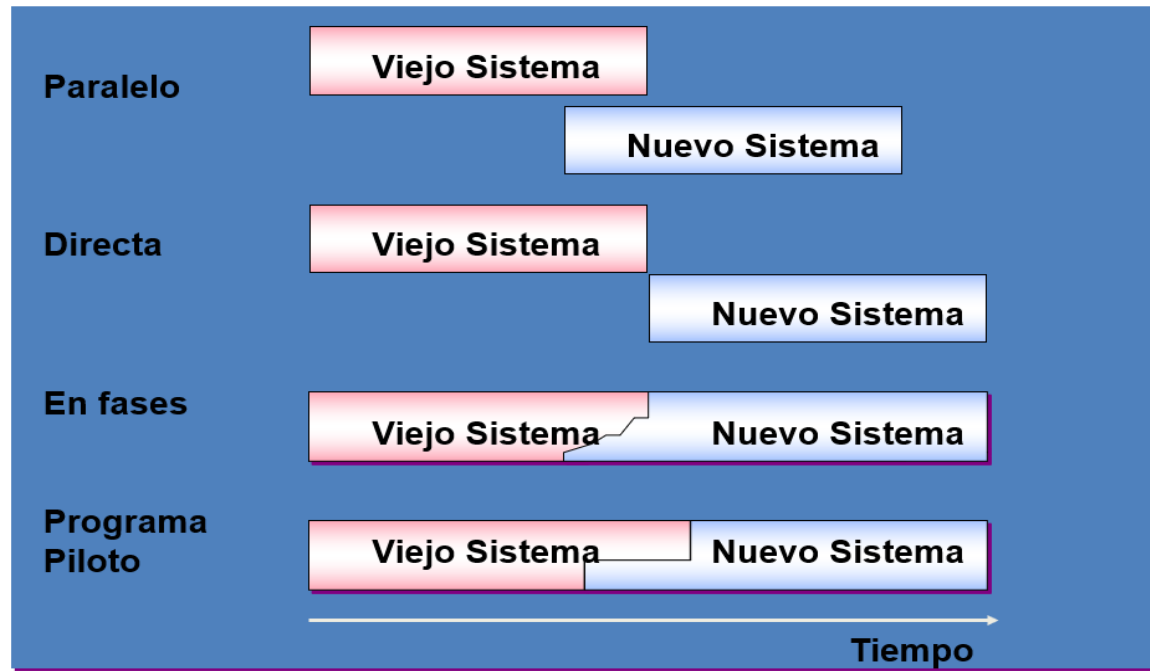
Estrategias de Despliegue

Cuando hablamos de estrategias de despliegue (también llamadas estrategias de implantación o implementación), nos referimos a la manera en que un nuevo sistema de información se pone en funcionamiento en una organización

ESTRATEGIAS CLÁSICAS

Estrategia	Descripción
Implantación Directa ("Big Bang")	Se apaga el sistema viejo y se enciende el nuevo en toda la organización en una fecha específica.
Implantación en Paralelo	El sistema nuevo y el antiguo funcionan simultáneamente por un tiempo. Es más seguro, pero más costoso. El tema clave es el período de overlap
Implantación por Fases / Áreas / Pilotos	Se implementa por etapas (por módulos, departamentos o procesos). Reduce riesgos y permite ajustes progresivos. Tener en cuenta los circuitos

Metodos de Conversion



ESTRATEGIAS MODERNAS

Estrategia	Descripción
Blue-Green Deployment	Se tienen dos entornos (blue y green). Se dirige el tráfico al nuevo entorno (green) cuando todo está probado. Permite reversión rápida.
Canary Deployment	Se lanza el nuevo sistema o versión a un pequeño subconjunto de usuarios. Si funciona bien, se amplía progresivamente.
Rolling Deployment	Se actualizan los servidores o instancias uno por uno. No hay necesidad de interrupción completa.
Feature Toggles (Interruptores)	Se implementa el sistema completo, pero ciertas funcionalidades están ocultas o desactivadas hasta ser probadas.
Shadow Deployment	El sistema nuevo procesa datos en segundo plano (sin afectar usuarios), para evaluar su comportamiento frente al sistema viejo.
A/B Testing	Se presentan dos versiones diferentes (A y B) a usuarios distintos, y se mide cuál funciona mejor. Se usa mucho en software web.

Comparativa de Objetivos

Estrategia	Riesgo	Costo	Flexibilidad	Tiempo de Despliegue
Big Bang	Alto	Bajo	Bajo	Corto
Paralelo	Bajo	Alto	Medio	Largo
Fases / Pilotos	Bajo	Medio	Alto	Largo
Blue-Green	Muy bajo	Medio	Muy alto	Medio
Canary	Bajo	Medio	Muy alto	Largo (escalonado)
Rolling	Bajo	Medio	Medio	Medio
Feature Toggles	Bajo	Medio	Muy alto	Rápido
Shadow	Muy bajo	Alto	Medio	Largo
A/B Testing	Bajo	Medio	Muy alto	Variable

USUARIOS - ROLES - PERFILES

En el contexto de sistemas de información (particularmente en bases de datos, ERPs, sistemas de gestión o plataformas de software empresarial), la configuración de perfiles, roles y usuarios es clave para garantizar la seguridad, trazabilidad y organización del acceso a los datos y funcionalidades del sistema.

Usuario

Un usuario es una cuenta individual que representa a una persona (real o técnica), aplicación o servicio que accede al sistema.

Tiene credenciales propias (usuario/contraseña).

Se le asignan **roles** y **perfiles** que determinan lo que puede ver o hacer.

 *Ejemplo:* Usuario jlopez puede ingresar a un ERP para cargar facturas, pero no para emitir reportes contables.

Rol

Un **rol** es un conjunto de permisos y privilegios agrupados bajo una etiqueta común, que se puede asignar a uno o varios usuarios.

Se basa en funciones o tareas.

Permite simplificar la administración de permisos.

Se puede actualizar una vez y afectar a todos los usuarios que lo tienen.

🌀 *Ejemplo:* Rol **facturador** incluye acceso a carga de ventas, emisión de comprobantes, pero no a reportes financieros.

Perfil

Un **perfil** es una configuración más detallada o personalizada, que puede incluir:

Parámetros de idioma, formato de fecha y moneda.

Límites de acceso por horarios o regiones.

Preferencias individuales de interfaz.

Seguridad adicional (doble autenticación, logs, etc.).

💡 *Diferencia clave:* Mientras el **rol** gestiona *lo que un usuario puede hacer*, el **perfil** gestiona *cómo y bajo qué condiciones* puede hacerlo.

🌀 *Ejemplo:* Dos usuarios con el mismo rol de **analista** pueden tener diferentes perfiles: uno ve reportes en español y otro en inglés.

ESTRATEGIA HABITUAL DE GESTIÓN DE USUARIOS, ROLES Y PERFILES

Una estrategia efectiva sigue el enfoque **RBAC (Role-Based Access Control)**, y se basa en los siguientes pasos:

1. Identificación de roles funcionales

Relevamiento de los procesos y funciones que se realizan en el sistema.

Agrupamiento de tareas similares bajo roles: administrador, analista, operador, gerente, etc.

2. Definición y documentación de roles

Cada rol debe tener:

Permisos específicos (lectura, escritura, eliminación).

Restricciones o límites.

Dependencias con otros roles (por ejemplo, quien aprueba necesita ver lo que otro genera).

3. Creación de usuarios

Se crean los usuarios con nombres únicos, asociando:

Su **rol principal**.

Uno o más **perfiles**, si es necesario.

Se documenta a qué áreas pertenecen y qué responsabilidades tienen.

4. Política de contraseñas y autenticación

Establecer reglas de complejidad, vencimiento y recuperación.

Implementar segundo factor si se requiere seguridad crítica.

5. Monitoreo y auditoría

Registrar accesos y actividades (log de auditoría).

Detectar accesos indebidos, cambios no autorizados o intentos de vulneración.

6. Revisión periódica de permisos

Hacer auditorías semestrales o anuales para revisar:

Usuarios inactivos.

Roles duplicados o sin uso.

Errores de asignación de permisos.

7. Gestión de bajas y cambios

Si un usuario cambia de sector o se desvincula, se deben:

Quitar sus accesos.

Reasignar funciones.

Registrar la baja en logs del sistema.

Buenas prácticas recomendadas

Principio de menor privilegio: dar solo los permisos estrictamente necesarios.

Separación de funciones: evitar que una persona pueda completar todo un circuito sin controles (por ejemplo, cargar y aprobar pagos).

Uso de roles temporales: para proyectos o reemplazos, con fecha de expiración.

Asignación automatizada: cuando se integra con Active Directory u otros sistemas de gestión de identidades.

¿Qué es una Cuenta de Servicio?

Una **cuenta de servicio** es una cuenta especial que **no representa a un usuario humano**, sino a **una aplicación, servicio, proceso automático o componente del sistema** que necesita autenticarse y operar en una infraestructura de TI.

Estas cuentas permiten a los servicios:

Acceder a bases de datos.

Ejecutar tareas programadas.

Interactuar entre servidores.

Consumir APIs.

Gestionar procesos en segundo plano.

En pocas palabras, las cuentas de servicio son "usuarios técnicos" que trabajan de manera automática y silenciosa para mantener el funcionamiento del sistema.

¿Qué son los parámetros en la implementación de sistemas?

Parámetros son **valores configurables** que definen **cómo se comporta el sistema** sin necesidad de modificar el código fuente.

Son "reglas" o "condiciones" que personalizan el software para adaptarlo al contexto y necesidades específicas de cada organización.

Tipos de Parámetros Comunes

Tipo de parámetro	Ejemplos
Operativos	Moneda base, idioma, formato de fecha.
Técnicos	Tiempos de timeout de sesión, número máximo de conexiones concurrentes.
Seguridad	Políticas de contraseña, intentos de login fallidos permitidos.
Funcionales	Políticas de facturación, cálculo de intereses, reglas de descuentos.
Integración	URLs de APIs, tokens de autenticación, direcciones IP autorizadas.
Control de procesos	Límite de registros por lote, frecuencia de ejecución de tareas automáticas.

5 - Implementación o Despliegue

- Acceso a datos y ambientes productivos
- Instalación on premises / local computers
- Acceso a recursos compartidos dentro y fuera organización
- Configuraciones técnicas - VPN et al

6 - Uso en Productivo - BAU (Business as usual)

- Lanzamiento - Go live
- Soporte de estabilización - usual 30 días

7 - Post - Producción

- Soporte
- Mantenimiento
- Nuevos Requerimientos
- Customizacion