Revisión de tecnologías

Revisión por cada capa

Capa 1 - Menos pasos, más propósito

Menos pasos, más propósito

Implicación para desarrollo

- Es una capa de resultados, más que de gestión de operaciones.
- Siento que es un reto más de diseño de GUI
- El front podría tener implicaciones implementando mecanismos de comunicación eventual.

Tecnologías que pueden ser necesario implementar

- SignalR
- Server Side Events
- React

Capa 2 - Tiempo como valor

Tiempo como valor

Focos temporales:

1. **Proactivo**. El sistema actúa antes de que alguien lo pida
2. **Sincrónico**. El sistema actúa en el instante correcto, sin retraso ni repetición
3. **Reactivo controlado**. El sistema espera a confirmar porque entiende que el tiempo tiene valor de validación

Implicación para desarrollo

- El sistema debe estar atento proactivamente. Es decir, saber qué debe ejecutar, dependiendod e variables:

  > Generar una alerta cuando falten 3 días para el vencimiento del contrato.

- El sistema debe reaccionar a ciertos eventos en línea.
- Otro caso es que esas reacciones puedan estar amarradas a lógica condicional. Como puede ser que se hayan registrado dos eventos para que se dispare:

  > La compra fue registrada **Y** el presupuesto está aprobado

- Todas estas acciones podrían estar prediseñadas en el código o **parametrizables** por los usuarios.

Tecnologías que pueden ser necesario implementar

- Event-driven Arquitecture (EDA)
- Sagas
- Scheduler / Cronos / Demons / Jobs o cualquiera de sus implementaciones
- Machine learning

Capa 3 - Repetición cero

Repetición cero

Evitar la duplicación de operaciones. En términos generales los flujos deben tener presente la información previamente digitada para que sea reutilizada por los pasos posteriores.

Evitar la duplicación de registros, se tenía muy controlado en Sinco ERP porque al ser un monolito, la capa de datos se compartía y se evitaban la mayoría de registros duplicados.

Para la arquitectura eventual, puede que tengamos un reto en duplicación de información entre microservicios.

Las exploraciones actuales por eventos, nos exige que la info necesaria por un dominio deba ser entregada en el evento de integración.

También debemos tener las suscripciones a los dominios que centralizan información para sincronizar efectivamente cambios en las fuentes.

Implicación para desarrollo

- Eventos de integración "gordos", para evitar los callbacks.
- Suscripciones a modificaciones de datos maestros por empresa.
- Documentación de eventos para evitar la duplicidad.
- Definición de cuál es el dominio productor de datos maestros.

Tecnologías que pueden ser necesario implementar

- Event-driven Arquitecture (EDA)
- Event Catalog
-

Capa 4 - Dependencias inteligentes

Dependencias inteligentes

Las dependencias las podemos modelar como flujos de procesos, ya sean orquestados o coreografiados.

Niveles de dependencias inteligentes:

1. **Notificable**. Solo informa al otro módulo (usuario)
2. **Accionable**. Ejecuta algo automáticamente
3. **Predictiva**. Actúa antes de que otra persona lo pida

Implicación para desarrollo

- Tener un mapa de dependencias se puede realizar con el mapa de eventos que muestran las relaciones entre los distintos dominios.
- En el nivel notificable, debemos implementar mecanismos de comunicación a terceros: correo, notificacione dentro del sistema, whatsapp, etc.

Tecnologías que pueden ser necesario implementar

- Event-driven Arquitecture (EDA)
- Event Catalog
- Sagas

Capa 5 - Anticipación funcional

Anticipación funcional

Reconocer los hábitos de los usuarios. Estos son los niveles de anticipación:

1. **Sugerencia preventiva**
2. **Acción automática con aviso**
3. **Acción autónoma**

Las anticipaciones son resultado de otras capas como :

- Capa 4 - Dependencias inteligentes
- Capa 9 - Aprendizaje continuo
- Capa 10 - Explicabilidad
- Capa 6 - Integración extendida

Implicación para desarrollo

Las implicaciones para desarrollo se desarrollarán en las capas dependientes. Esta capa solo invoca las demás para presentar la anticipación.

Tecnologías que pueden ser necesario implementar

- No aplican

Capa 6 - Integración extendida

Integración extendida

Implicación para desarrollo

- Debemos construir capas de integración con sistemas externos, como por ejemplo:
  - API Rest
  - Webhooks
  - Comunicaciones: correo electrónico, whatsapp.

    - Datos: Excel, csv, Google sheets, datalakes, etc.
- Importación y exportación de información
- Guardar documentos en diferentes blob storages
- Documentar todas las API para consumos de terceros

Tecnologías que pueden ser necesario implementar

- Capa de integración:
    - Rest API
    - Webhooks
    - GraphQL
    - MCP servers
- Soluciones de correo electrónico integrado:
    - Servidores SMTP
    - Servicios de correo electrónico de terceros. Ej: Sendgrid, Mailgun, etc.
- Importación de datos:
    - csv
    - txt
    - Hojas de cálculo
- Exportación de datos:
    - csv
    - Hojas de cálculo
- Creación de plantillas para documentos
    - Mustache
    - Software para generación de pdf. Ej: iTextSharp
- Documentación de servicios:
    - Open API
    - Async API
- Integración con blob storage
    - Azure blob storage
    - S3 Buckets

Capa 7 - Comunicación contextual

Comunicación contextual

Debemos definir los canales de comunicación que vamos a utilizar para darle información a nuestros usuarios:

- Notificación contextual
- Panel de comunicaciones / dashboard
- Correo
- Whatsapp
- Notificación en la APP

Implicación para desarrollo

- Orquestar cada canal que se determine en la aplicación.
- Habilitar capa interactiva para que se permitan acciones por parte del usuario. Ej: Aprobaciones

Tecnologías que pueden ser necesario implementar

- Desarrollo mobile
- Flujos por whatsapp
- Soluciones de correo electrónico integrado:
    - Servidores SMTP
    - Servicios de correo electrónico de terceros. Ej: Sendgrid, Mailgun, etc.
- Scheduler para definir los momentos de comunicación

Capa 8 - Adaptación dinámica

Adaptación dinámica

Implicación para desarrollo

- Definición de roles y accesos
- Tour guiado para nuevos usuarios
- Front adaptativo según el usuario

Tecnologías que pueden ser necesario implementar

- Usetiful u otras herramientas que se superponen a la aplicación para moderar las interacciones del usuario.
- Investigar como hacer experiencias adaptativas del front amarrada a roles y permisos
- Capa de permisos y accesos
  - RBAC

Capa 9 - Aprendizaje continuo

Aprendizaje continuo

El sistema debe aprender a los siguientes niveles:

```
1. Usuario individual
2. Rol
3. Empresa
4. Todas las empresas
```

Implicación para desarrollo

- Debemos observar y guardar las interacciones de los usuarios, para encontrar patrones y aprender
- Debemos analizar periódicamente las interacciones.
- Debe haber un sistema para entregarle los aprendizajes a los usuarios, consultores y desarrolladores.
- Crear mecanismos para modificar los flujos dependiendo de los aprendizajes

Tecnologías que pueden ser necesario implementar

- Logs de las acciones generadas por los usuarios
- Capturar comportamientos recurrentes de la interfaz gráfica: Clarity o similares.
- IA a partir de los logs y todas las fuentes de información de comportamiento
  - RAG

Capa 10 - Explicabilidad

Explicabilidad

Niveles de explicación:

1. Contextual. Dentro del flujo sin interrumpir al usuario.
2. Explicativo. Cunado el usuario pasa el cursor o pide detalles.
3. Auditoría/Reportes. En un panel donde se vea la trazabilidad.

Implicación para desarrollo

- Notificaciones contextuales en el front.
- Diseño de dashboards.

Tecnologías que pueden ser necesario implementar

- SignalR

Capa 11 - Transparencia operativa

Transparencia operativa

Formatos de visibilidad:

1. Trazabilidad en tiempo real. Línea temporal con eventos del flujo
2. Historial detallado. Registro completo y accesible por cada documento del módulo.
3. Panel global de operaciones. Dashboards donde los líderes puedan ver la salud del sistema.

Implicación para desarrollo

- Capturar eventos con la mayor cantidad de información no estructurada posible.
- Procesar altos volúmenes de registros para creación de los productos de información.
- Múltiples fuentes y tipos de datos pueden ser necesarios para brindar la información completa.

Tecnologías que pueden ser necesario implementar

- OpenTelemetry

- Registro de logs: Elasticsearch
- Tableros de observabilidad Kibana, Prometheus

Capa 12 - Supervisión ética

Supervisión ética

Niveles de supervisión:

1. **Autónoma**. Sin riesgo -> el sistema actúa solo.
2. **Supervisada**. Riesgo medio -> requiere confirmación.
3. **Restringida**. Riesgo alto -> solo humano puede decidir.

Mecanismos de control:

1. Conformación por roles o permisos
2. Panel de revisiones y confirmaciones
3. Alertas de intervención
4. Reglas de reversión

Implicación para desarrollo

- Las acciones autónomas deben estar programadas por desarrollo o parametrizables por parte de los usuarios.
- Las parametrizaciones pueden estar expresadas en distintos tipos de límites, montos, tiempos, etc.
- Se pueden tener múltiples niveles de aprobación y podrían ser escalonados.

Tecnologías que pueden ser necesario implementar

- Control de accesos basados en roles RBAC
- Modelador de aprobaciones dinámicas.

!

Capa 14 - Radar de negocio

Radar de negocio

Tipos de conectividad predictiva:

1. **Predictiva**. Detecta tendencias y advierte antes que ocurra.
2. **Sincrónica**. Conecta módulos para evitar desalineaciones.
3. **Proactiva**. Actúa antes del usuario, con confirmación.

Ciclo de predicción:

1. Dato detectado (Indicador afectado).
2. Patrón identificado.
3. Riesgo proyectado
4. Acción sugerida o ejecutada
5. Validación del resultado

Implicación para desarrollo

- Sistema de monitoreo activo de métricas de control.
- Configuración de tableros para construir el ciclo de predicción

Tecnologías que pueden ser necesario implementar

- Elasticsearch
- Prometheus

Capa 15 - Orquestación contextual

Orquestación contextual

Implicación para desarrollo

- Adaptación de la interfaz gráfica dependiendo: qué rol tiene el usuario, en qué módulo está y qué evento temporal está sucediendo en el momento.

Tecnologías que pueden ser necesario implementar

- React