

ADR - Integración de microfrontends en aplicaciones

ADR -UX-UI-MF-HOST-INTEGRATION

Estado	VIGENTE			
Partes interesadas	AREA GOBIERNO TECNOLOGICO			
TAGS	UX-UI FRONT-ARCHITECTURE			
Tomada el	5 jul 2024			
Responsable	LUIS MARTINEZ FONTIVEROS			
Solución Afectada	TODAS			
Documentación relacionada	Fichero		Modificado	
	Archivo PDF Normativa-v8-20240705_111105.pdf		05/07/2024 by LUIS MARTINEZ FONTIVEROS	
	Fichero App-shell.excalidraw		05/07/2024 by LUIS MARTINEZ FONTIVEROS	
	Archivo PNG image-2024-7-5_11-13-38.png		05/07/2024 by LUIS MARTINEZ FONTIVEROS	
	Descargar todo			
	Responsable	Aprobado por	Consultados	Informados
	LUIS MARTINEZ FONTIVEROS	LUIS MARTINEZ FONTIVEROS CARLOS GONZALEZ SANTIAGO		AREA GOBIERNO TECNOLOGICO DESARROLLO SOFTWARE ASISTENCIAL (NTTDATA) DESARROLLO SOFTWARE ECONOMICO-FINANCIERO (NTTDATA) DESARROLLO SOFTWARE RRHH (AYESA)

Asunto a Decidir

Arquitectura de desarrollo e integración de los microfrontend dentro de las Aplicaciones del SAS

Objetivos:

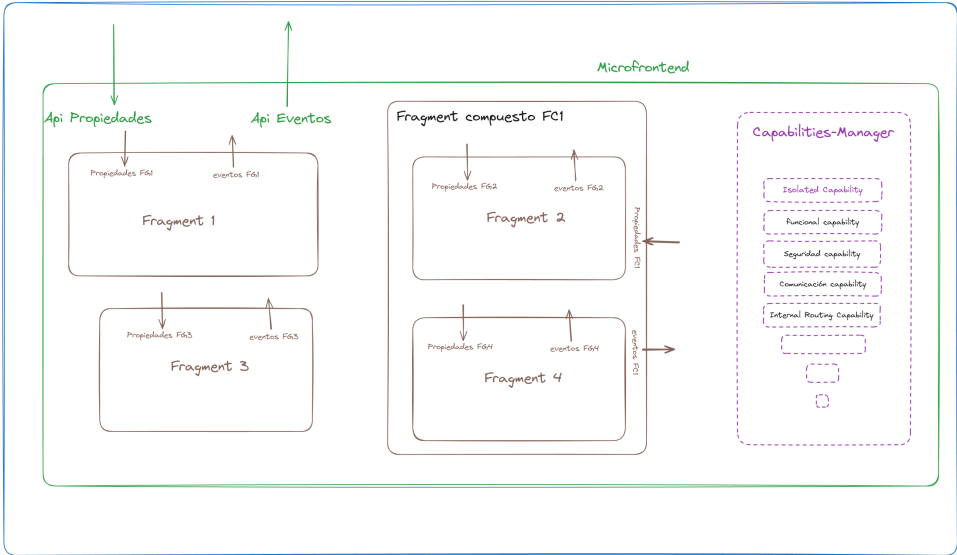
- Simplicidad
- Compatibilidad tecnológica
- Minimización del impacto en las aplicaciones integradoras

Identificación de la decisión

La arquitectura se compone de 4 elementos:

- **Host:** Elemento integrador que hará uso del microfrontend
- **Microfrontend:** Elemento encapsulador que expondrá una api de integración con los fragments de los que está compuesto
- **Fragment:** Unidad funcional frontal que tiene una única responsabilidad funcional
- **Capability:** Elemento funcional (general o específica) que tiene una única responsabilidad

Host (Aplicación en la que integrar el Microfrontal)



Contexto	Requerimientos de la solución: i Alta capacidad de reutilización Cómo organización quiero una solución reutilizable y que sólo requiera de configuración diferente para integrar aplicaciones nuevas, aplicaciones legacy o microfrontales, para evitar reimplementar lo mismo de varias formas y mejorar el mantenimiento. i Estandarización en el modo de integrarse entre aplicaciones del front. Cómo organización quiero una solución estándar de integración entre aplicaciones en el frontal, de modo que los equipos no tengan que reinventar la solución para cada integración definida, para reducir los costos de implementación y mantenimiento. i Composición Cómo arquitecto, quiero que la solución no se encuentre limitada en niveles de composición para poder definir modelos de integración complejos de modo que una aplicación pueda considerarse: ◦ Microfrontend cuando se integra en otra aplicación. ◦ Microfrontend cuando se integra en una legacy via Microfrontend. ◦ Aplicación cuando se instancia por si sola. i Navegabilidad Cómo arquitecto, quiero dar al usuario la capacidad de navegar entre diferentes espacios del frontal, de forma transparente a si los elementos que se articulan en la aplicación, son aplicaciones externas o microfrontales integrados, para tener una experiencia de usuario cohesionada y homogénea. i Estilos Como arquitecto, quiero dar al usuario una sensación de homogeneidad en la apariencia y en la experiencia de usuario, para que el usuario no tenga mental-breaks cuando está trabajando con diferentes aplicaciones y reducir la curva de aprendizaje.
----------	--

Alternativas Consideradas

Build Time integration

La integración directa en build time, consiste en la definición del microfrontend cómo dependencia en tiempo de implementación de la solución, para ello un microfrontend se considerará como una librería (componente) que se importa dentro de la solución a implementar.

Pros
Facilidad en el proceso de integración: El proceso de integración es determinista, al integrarse como dependencia
Control de errores funcionales: Al ser una integración en build time, las garantías de validación funcional se transfieren al entregable.

Contras	Mitigación
Las dependencias transitivas del microfrontend pueden generar conflicto con el Host	Depende de la situación: • El host debe actualizar la librerías conflictivas. • En caso de que el host tenga una versión superior de la dependencia se debe notificar al responsable del microfrontend para solicitar la actualización • En caso de que no se posible eliminar el conflicto, se deberá usar el modo de integración embebed
Cada modificación funcional que se produzca en los microfrontend integrados requiere de una reimplementación, versionado y despliegue de la aplicación que hospeda.	En la construcción de los microfrontend, si se sigue Semantic version se reduce el impacto en el host con las entregas

Evaluación de Objetivo

Requisito	OK/NOK	Detalle
Simplicidad	OK	Uso a través de la dependencia que con una configuración determinada por el microfrontend se podrá realizar la integración
Compatibilidad tecnológica	A MEDIAS	Siempre que el host y el microfrontend utilicen una misma librería con diferentes versiones se deberá realizar el plan de mitigación

Minimización del impacto en las aplicaciones integradoras	A MEDIAS	- La simplicidad y que el host no tenga que realizar ningún desarrollo específico para la integración. - En caso de conflicto puede haber un impacto difícil de cuantificar a priori
---	----------	---

Embebed integration

La integración se realiza a través de la url donde esté desplegado el microfrontend o la aplicación que se quiera embeber. De esta forma cuando se accede a la url, el microfrontend o aplicación, actúan como una aplicación independiente, y cuando se embeben actúan como un microfrontend integrado.

Para poder llevar a cabo esta integración, se deben cumplir los siguientes requisitos imprescindibles:

- El microfrontend o aplicación, al menos en su capa más exterior debe estar desarrollado con [lit](#).
- El microfrontend o aplicación, debe tener instalado el capability manager ([@sas/lib-stic-capabilities](#)) en su componente más exterior.
- En el host, se debe hacer uso del componente [@sas/wc-stic-isolated-microfrontend](#) que creará un contexto independiente donde se ejecutará el microfrontend dentro del host.

Si se desea una integración homogénea estilística entre el host y el microfrontend se deben cumplir los siguientes requisitos imprescindibles:

- El host debe estar desarrollando aplicando el [@sas/wc-stic-theme](#)
- El microfrontend o aplicación debe estar desarrollando aplicando el [@sas/wc-stic-theme](#)

Pros
Bajo impacto en la integración. Se debe usar un componente ya proporcionado e indicarle la URL de integración.
Aislamiento tecnológico, lo que aumenta las posibilidades de integración sobre todo con las aplicaciones legacy

Contras	Mitigación
Al aislarse en un contexto independiente , puede ocurrir que se descarguen dos veces los mismos recursos y no se podrá reutilizar los recursos previamente obtenidos por el host	Hacer uso de las mínimas librerías necesarias y que estén bundelizadas y minificadas

Al requerirse desarrollarse dentro de un contexto tecnológico acotado (web-component y lit) en los proyecto legacy (tanto host como aplicaciones legacy a embeber) puede requerir algún ajuste previo inicial para poder utilizarlo	N/A
---	-----

Evaluación de Objetivo

Requisito	OK/NOK	Detalle
Simplicidad	OK	Uso a través de la dependencia que con una configuración determinada por el microfrontend se podrá realizar la integración
Compatibilidad tecnológica	OK	Al estar en un contexto aislado no habrá conflicto tecnológico entre el host y el microfrontend
Minimización del impacto en las aplicaciones integradoras	A MEDIAS	- La simplicidad y que el host no tenga que realizar ningún desarrollo específico para la integración. - Evitar conflictos entre el host y el microfrontend - Tanto el host como el microfrontend deben estar dentro del contexto tecnológico marcado por la arquitectura de referencia del front para poder funcionar sin impacto

Decisión Tomada

Siempre que se pueda se optará por la opción de integración en build time. En caso de conflicto tecnológico o justificación de un aislamiento, se podrá utilizar la integración embebida.

Consecuencias

- Las aplicaciones host que incorporen el microfrontal deberán evaluar su stack tecnológico de front-end. En función de sus capacidades tecnológicas, pueden presentarse dos escenarios principales:
 - La aplicación está preparada tecnológicamente, y ha cubierto las Normativas de Desarrollo TIC para la interfaz de usuario adecuadamente. En este caso la aplicación host incorporará en build time el microfrontal a integrar, y deberá recompilar, empaquetar y desplegar.
 - La aplicación no está preparada tecnológicamente o no ha cubierto las Normativas de Desarrollo TIC para la interfaz de usuario adecuadamente. En este caso la aplicación host incorporará en build el microfrontal a integrar (que será especial, del tipo "Isolated-Microfront" que incorpora un lframe internamente para garantizar el aislamiento), y deberá recompilar, empaquetar y desplegar.

Estado Actual



Describe brevemente la decisión arquitectónica tomada.

Siempre que se pueda se optará por la opción de integración en build time. En caso de conflicto tecnológico o justificación de un aislamiento, se podrá utilizar la integración embebida.