

The screenshot displays the Swagger Editor interface. At the top, there are four tabs: 'REST(BOM Expanded)', 'REST(ISO Expanded)', 'Async(BOM Expanded)', and 'Async(ISO Expanded)'. The main editor area is split into two panes. The left pane shows the OpenAPI JSON definition for a 'Savings Account' API, including details like title, description, contact information, servers, and a list of paths. The right pane provides a visual representation of the API, showing the title 'Savings Account' with version indicators '12.0.0' and 'OAS 3.0', a description, a link to the BIAN website, and a list of endpoints under the 'CR - SavingsAccountFacility' section. The endpoints listed are: a POST endpoint for '/SavingsAccount/Initiate', a PUT endpoint for '/SavingsAccount/{savingsaccountid}/Update', and another PUT endpoint for '/SavingsAccount/{savingsaccountid}/Control'.

La transformación digital en la banca ha generado la necesidad de estandarizar las **APIs financieras**, facilitando la interoperabilidad y optimización de servicios. En este contexto, **BIAN Semantic API** se ha convertido en el marco ideal para diseñar **APIs bancarias modulares, escalables y alineadas con estándares globales como ISO 20022**.

## ¿Qué es BIAN Semantic API?

BIAN (Banking Industry Architecture Network) define un modelo de **Service Domains** que permite estructurar las operaciones bancarias de manera uniforme. Las **BIAN Semantic APIs** utilizan estos dominios para diseñar servicios interoperables, organizados en **Behaviour Qualifiers** y **Action Terms**, asegurando consistencia y facilidad de integración.

## ¿Por qué adoptar BIAN Semantic API?

- **Estandarización:** Define un lenguaje común para integrar servicios bancarios.
- **Interoperabilidad:** Facilita la conexión entre bancos, fintechs y reguladores.
- **Reducción de costos:** Disminuye los tiempos de desarrollo e implementación.
- **Escalabilidad:** Permite evolucionar los servicios sin afectar la arquitectura base.
- **Cumplimiento con ISO 20022:** Asegura compatibilidad con el estándar global de mensajería financiera.

## Ejemplo de una URL en BIAN Semantic API

Si queremos recuperar el saldo de una cuenta de ahorros, la estructura recomendada sería:

```
/v1/savings-account/balance/{accountId}/retrieve
```

Este enfoque garantiza claridad, modularidad y alineación con estándares internacionales.

## Aprende más sobre BIAN Semantic API

Para profundizar en la implementación de **BIAN Semantic API**, solicita una sesión con un experto o revisa nuestros cursos y consultorías especializadas en **ArquitecturaBANK Consulting**.

□ **Cursos disponibles:** <https://arquitecturabank.com/courses/>

□ **Consultorías especializadas:** <https://arquitecturabank.com/servicios-consultoria-bian/>

□ **Optimiza la arquitectura de tus APIs bancarias con BIAN. ¡Da el siguiente paso en la transformación digital!**

#BIAN #APIs #ISO20022 #Fintech #BancaDigital #ArquitecturaBancaria

### Julio Pari (IT Architect BIAN)



Especialista BIAN Semantic API | Gobierno de Integración | IBM Integration CP4I | IBM API Connect 10 | IBM ACE | IBM DataPower | OpenShift | Azure | AWS. Cualquier consulta envíame un mensaje a: [info@arquitecturabank.com](mailto:info@arquitecturabank.com) o sino a través de LinkedIn: <https://www.linkedin.com/in/juliopari/>