

Guía de ejercicios de Arquitectura: Soluciones

Franco Bulgarelli

Versión 2.0

Octubre 2023

A continuación encontrarás una serie de soluciones propuestas para [la guía de ejercicios de Arquitectura](#). Las mismas tan sólo esbozan, a alto nivel, algunas alternativas posibles, y no descartan otras opciones.

EntregaYaYaYa	1
Aclaraciones	1
Opción 1	2
Opción 2	2
MicroPC IDE	2
Aclaraciones	2
Rekomendashi, Fase 2	4
Aclaraciones	4
Shera Enterprise	4
Aclaraciones	5
Cyberfriday	5
Librerías Bustos Domecq	6
A.F. Analytics	7

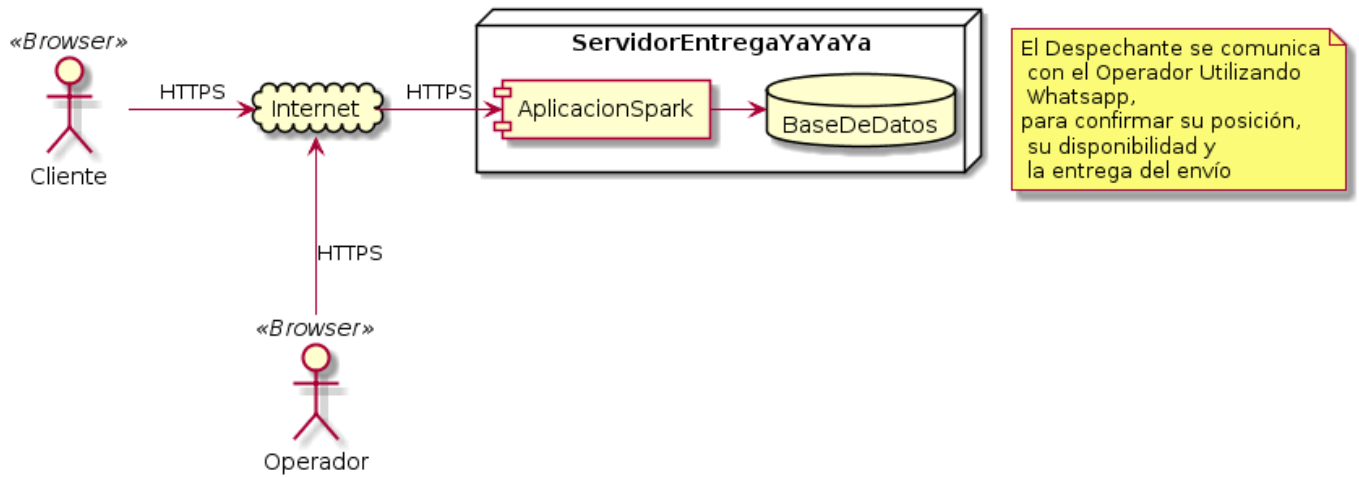
EntregaYaYaYa

Aclaraciones

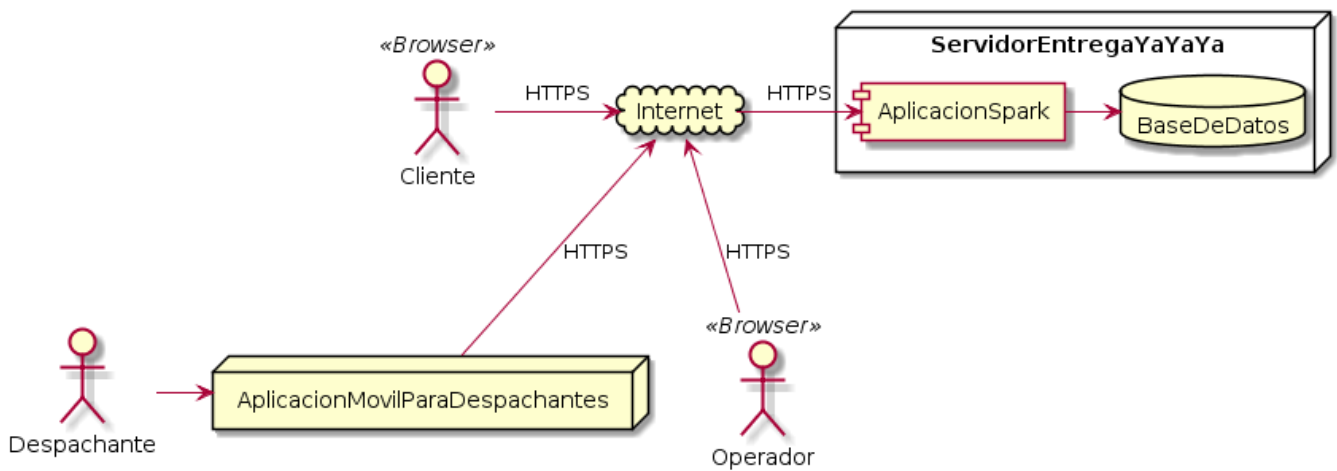
- Este ejercicio no sólo está para discutir arquitecturas clásicas de software sino para ver distintos actores en el sistema completo, distintas necesidades e integraciones y problemas que pueden resolverse sin una línea de código (por ejemplo, podríamos contactarnos y saber la ubicación de nuestros despachantes por whastapp, y resolver la confirmación también por este medio, de forma semi-manual)
- Hay que discutir algunos aspectos de la logística: de alguna forma el cliente y el despachante deben confirmar la recepción del pedido, de alguna forma las personas operadoras deben poder enterarse de que hay un pedido nuevo, y de alguna forma debe poder contactar y saber dónde está quien lo despachó.
- También hay que discutir los distintos Cloud Providers y Bare Metal
- Más allá de eso, la arquitectura es bastante clásica: Cliente Web Liviano + Servidor Web + Base de Datos.

Se podría pensar también como alternativa un cliente web pesado o una aplicación móvil. La persona operadora no necesitará una aplicación móvil, para le usuarie es deseable, pero para le despachante es vital.

Opción 1



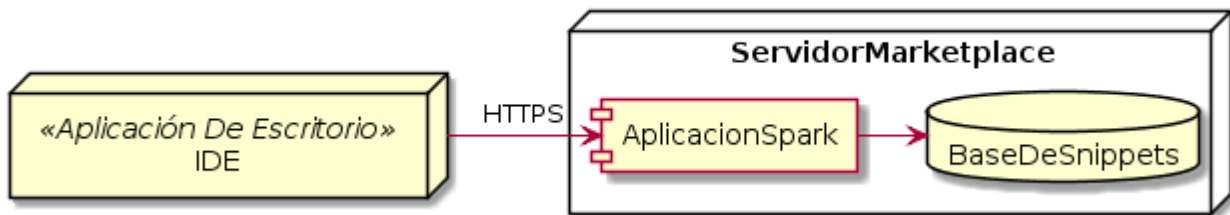
Opción 2



MicroPC IDE

Aclaraciones

- Los componentes web para editar código son cada vez mejores, pero suelen ser muy limitados comparados contra sus contrapartes nativas
- El acceso a los periféricos está limitado desde un navegador Web
- Programar desde un celular no es el principal caso de uso



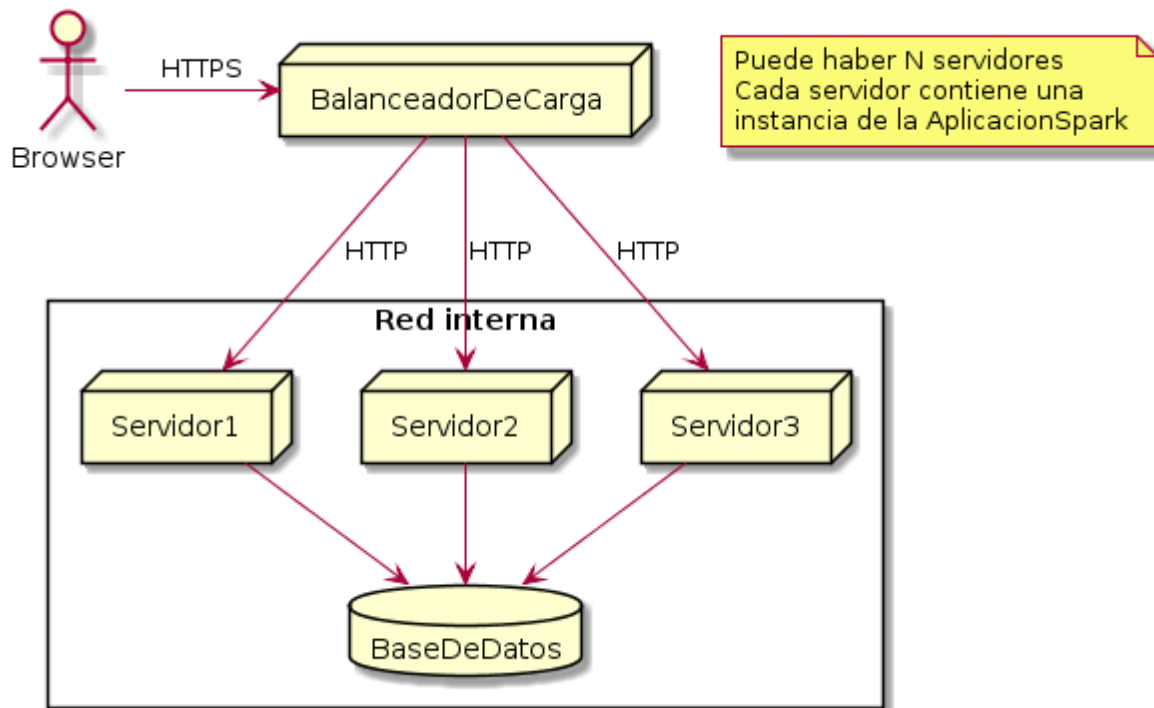
Notas:

- Aunque la interfaz es Desktop, la arquitectura es Web
- La aplicación Spark expone un API (por ejemplo, un API REST)

Rekomendashi, Fase 2

Aclaraciones

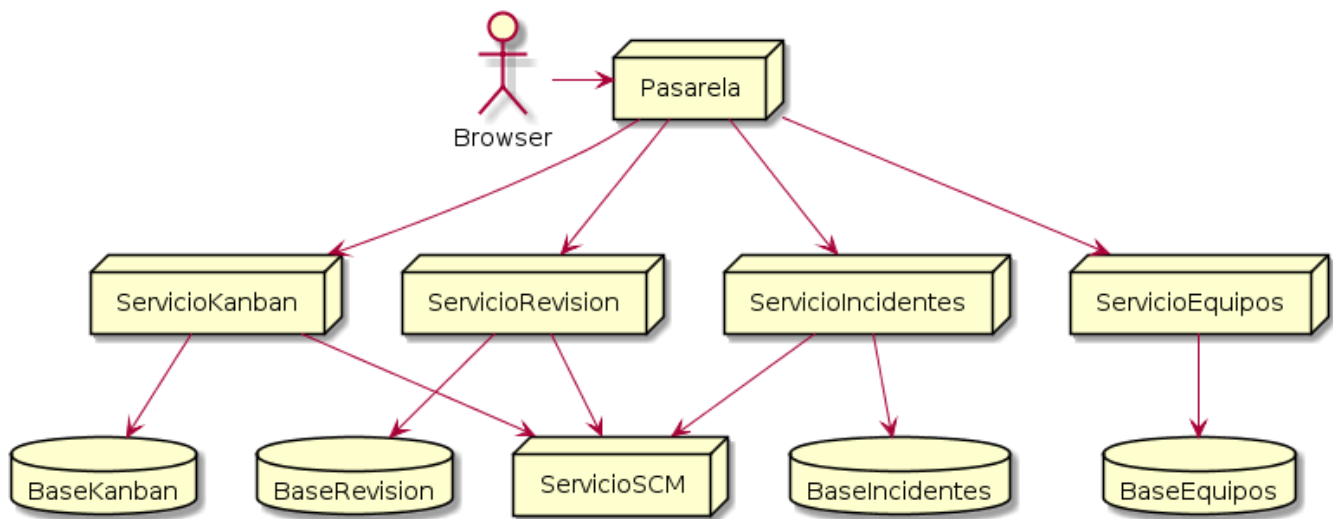
Este ejercicio apunta a introducir un load balancer y una CDN, a extraer la base a un servidor aparte, y a empezar a escalar horizontalmente



Notas:

- Dentro de la red interna no es necesario usar HTTPS

Shera Enterprise



Aclaraciones

- No todos los servicios tienen que ser stateful, y que es deseable que no compartan almacenamiento pero no está prohibido, y que todo está dentro de una red interna, y que la comunicación es típicamente sincrónica y mediante HTTP, pero tampoco es necesario
- Tener en cuenta que probablemente haya elementos comunes entre los dominios de cada servicio: por ejemplo, todos probablemente tengan que interactuar con el concepto de tarea, proyecto, empresa/cliente, etc.
- Problemas abiertos a los que aún hay que responder: ¿Cómo comunico a los sistemas? ¿Cómo mantengo la consistencia entre ellos? ¿Qué tecnologías utilizo? ¿Qué hago con el estado? ¿Cómo manejo la transaccionalidad?
- Lo partimos en múltiples servicios o microservicios. Surgen ventajas y desventajas: performance vs organización, complejidad vs agilidad organizacional.

Cyberfriday

Consideraciones:

1. ¿Por qué puede estar tardando mucho? Si está normalizado, es una query que equivale a hacer un group by de la tabla de ventas, agrupando por producto, filtrando por fecha y categoría, haciendo un top N.
2. Encima, si consideramos que la consulta puede ser realizada en el nivel 2, y que eso se tiene que traer también los niveles inferiores, hay que hacer un join con la tabla de categorías, y con esta, un autojoin.
3. Opciones:
 - a. Desnormalizar la categoría nivel 2 en el producto, esto es necesario para posibilitar las consultas arbitrarias por categoría.
 - i. Si hubiera que habilitar búsqueda sobre cualquier categoría, se podrían usar arrays.
 - b. Complementario a esto, tenemos que agilizar la selección y agrupamiento.
 - i. Cacheando. Cuidado que va a tener una hora de error, y el primer usuario va a pagar un costo alto. Esto último se puede paliar dándole temporalmente un valor viejo si la entrada expiró.
 - ii. Otra es tener un contador global y reiniciar cada tanto.
 - iii. Otra es tener una tabla de menor tamaño, de la que borraremos entradas periódicamente.

Librerías Bustos Domecq

Más allá de plantear una nueva arquitectura, esta situación evidencia varias falencias del sistema:

- Probablemente no sea posible optimizar al módulo de estadísticas como para que se adapte a la velocidad de producción de eventos del módulo de ventas, pero valdría la pena revisar su implementación en búsqueda de cuellos de botella en la inserción de los datos.
- Es inaceptable, tanto desde un punto de vista de rendimiento como de seguridad, que un servicio bloquee a otro por tardar demasiado en responder. Toda llamada sincrónica a un servicio (ya sea interno o externo, aunque es aún más importante en los externos) debería tener un timeout razonable asociado¹. De todas formas tener cuidado: aunque la operación de timeout, no significa necesariamente que no se haya realizado.
- Es igualmente inaceptable que una operación transaccional fundamental del sistema como lo es una venta falle por una estadística que no se puede generar. El sistema debería manejar este error, marcando de alguna forma a la venta para generar estadísticas más tarde. Pero, ¡cuidado!: si simplemente ignoráramos la excepción estaríamos perdiendo información, lo cual tampoco es deseable. Esto de que la operación si bien es importante, no es fundamental, es análogo a lo que sucede con observers.
- Si los logs se están saturando, eso probablemente signifique que el sistema no cuenta con una política adecuada de rotación de los mismos, que descarte los más viejos una vez que rebalse

¹ En Java, por ejemplo, es posible fijar un timeout a un cliente Jersey mediante la propiedad `jersey.config.client.connectTimeout`

un cierto porcentaje ocupado del sistema de archivos. Otra cosa a revisar: ¿quizás el log está en DEBUG?

Dicho esto, es claro que tenemos dos servicios, P y C, P genera datos que C consume, mediante un API HTTP. ¿De qué formas podríamos hacer que la información viaje de P a C? P puede notificar a C, pero eso significa que si C va muy lento, P se queda colgado, o que si C se cae, P no puede procesar. O bien, en ambos casos se pueden perder mensajes. También puede ser al revés: C le pega a P cuando está listo para procesar un mensaje. Pero ahí sí P está caído o muy demandado, C queda ocioso.

Variantes: Monitorear y supervisar los servidores para minimizar el downtime. Usar una cola de mensajes.



A.F. Analytics

Los ejercicios anteriores están fundamentalmente orientados a proponer una arquitectura en base a los requerimientos funcionales y no funcionales. Sin embargo, el objetivo de este problema es diferente: analizar y discutir cómo éstos condicionan a la arquitectura, y cómo ésta puede tener un impacto social negativo, ya sea por acción u omisión. En otras palabras, no es que no se pueda plantear, por ejemplo, un diagrama de despliegue y estrategias tecnológicas para resolver los problemas propuestos (de hecho, invitamos a hacerlo), sino que en este caso nos interesa poder razonar críticamente sobre la arquitectura y cómo las decisiones que tomemos pueden afectar a la vida de las personas.

Como se observará, éste proceso es importante, dado que de no realizarlo de forma consciente probablemente terminemos contribuyendo a la concentración del poder de los grupos sociales privilegiados y perjudicando a aquellos marginados. En otras palabras, los sistemas en general y software en particular tienen la capacidad de reforzar estereotipos y provocar actos de discriminación (ya sea por motivos étnicos, de género, religión, edad, clase social, etc), pero también de contrarrestarlos. El poder de tomar decisiones e inclinar la balanza en uno u otro sentido también está en manos de quien diseña.

Al igual que en los ejercicios anteriores, no existen respuestas correctas o incorrectas a este problema, pero sí respuestas con mayor o menor nivel de fundamento. A continuación enumeramos brevemente algunas observaciones críticas sobre los requerimientos, esbozadas en forma de pregunta abierta. No las presentamos jerarquizadas por nivel de importancia, sino simplemente por orden de aparición en el texto.

- (...) eliminar completamente los trámites tradicionales en papel: ¿es realmente posible, o aún deseable, eliminar los trámites en papel? ¿Qué grado de confianza le dará a varias generaciones para las que el papel ha sido el medio principal de gestión de formularios durante toda su vida? ¿Todas las personas estarán en condiciones de realizar la totalidad de las gestiones de forma digital? ¿Estamos incurriendo en un sesgo capacitista?
- (...) el 100% de las gestiones (...) serán virtuales: ¿tiene sentido eliminar la presencialidad en absolutamente todos los trámites, incluso en aquellos que pueden ser críticos para la salud y seguridad de las personas? ¿Qué riesgos puede traer la suplantación de identidad en gestiones como pasaportes o tenencia de armas? ¿Qué recaudos podemos tomar en aquellos en que el riesgo es bajo?
- (...) *el ahorro que significará en infraestructura edilicia, papel y mano de obra*: las ventajas de digitalizar procesos en papel son evidentes, pero es necesario ponderar el costo que significará no sólo el desarrollo sino el mantenimiento de un sistema de tal envergadura, además de la capacitación de múltiples niveles del estado.
- Más allá de algunas voces aisladas (...): ¿cuáles son esas voces? ¿qué plantean? ¿presentan argumentos razonables que puedan mejorar nuestro entendimiento del sistema? Es importante no caer en argumentos de índole ad populum.
- El proyecto se desarrollará a lo largo de 18 meses (...): ¿resulta razonable pensar que una herramienta de tal alcance pueda realizarse en tan sólo 18 meses?
- (...) nuestra consultora, 2Diseños, ha sido recomendada por el Ministerio de Futuro Digital y seleccionada para su construcción: ¿es suficiente que una empresa sea considerada meritoria por un ministerio para que sea la encargada de realizar esta tarea? ¿Ha habido un debido (teniendo en cuenta el presupuesto) proceso de licitación? *Dado que el software será construido por una organización privada, ¿qué requisitos debería tener el software producido para que el estado pueda disponer completamente de éste?*

- ¿Te llama la atención algo de la foto? ¿Es un equipo diverso? ¿Podrá un equipo así abarcar la complejidad del software y diversidad de usuaries? ¿Qué sesgos puede presentar?
- Signup y login, el cual se resolverá utilizando documento de identidad o pasaporte (*numérico*), ¿Por qué debe ser numérico? ¿Podría eso traer problemas, sobre todo teniendo en cuenta que los requisitos implican que deben poder acceder también personas extranjeras?
- (...) *sexo (hombre/mujer)*. ¿Tiene sentido requerir esta información para poder registrarse? ¿Es relevante? ¿Por qué se plantea en términos de sexo y no de género? ¿Qué poblaciones estamos excluyendo al presentar una clasificación binaria?
- *Panel de control administrativo, en el cual se podrán obtener métricas, desagregadas por rango etario*: ¿es el eje etario el único relevante? Es importante tener presente que no es posible realizar análisis ni tomar decisiones sobre lo que no se mide. Por ejemplo, no discriminar estadísticas según género impedirá hacer análisis en este eje, así como no desagregar informes según ciudad o provincia impedirá aplicar una perspectiva federal.
- *Generación de un código QR de acceso directo a ciertos trámites, como formularios de aduana y declaraciones juradas*: nuevamente al requerir que los enlaces se compartan (exclusivamente) mediante códigos QR estamos forzando a que todas las personas cuenten con un celular con la tecnología capaz de procesarlos
- Por este motivo se está realizando una evaluación de costos de proveedores de infraestructura como Amazon, Azure y Google Cloud, entre otros: todas estas alternativas son conocidas por su robustez, escalabilidad y seguridad, pero, ¿podemos asegurar el control sobre los datos y software si el mismo está instalado en proveedores (potencialmente) en el extranjero o bajo una jurisdicción extranjera? En otras palabras, ¿cómo podemos asegurar la soberanía tecnológica en un sistema de tales características?
- Cada habitante en edad adulta deberá contar con la aplicación instalada, dado que los trámites son privados, seguros, personales e intransferibles. ¿Por qué sólo es accesible a través de celulares? ¿Qué problemas tiene esa decisión? ¿Cuenta toda persona con un dispositivo celular? ¿Es su obligación? Y aún en caso de poder contar con éste, ¿es capaz de usarlo? Más aún: ¿es toda persona capaz de gestionar sus propios trámites?
- ¿Qué problemas relativos al consentimiento y la protección de datos personales presenta la propuesta? ¿Qué sesgos pueden presentar los modelos generados? ¿Qué riesgos tiene la construcción y uso de tales modelos? ¿Podemos estar incurriendo en un caso de extractivismo tecnológico?