

1. Definición y Análisis de Temas

1.1 Frontend: Svelte con TypeScript (TS)

Definición:

Svelte es un framework frontend que compila componentes en JavaScript puro durante el build time, eliminando la necesidad de un runtime en el navegador. TypeScript añade tipado estático para mejorar la calidad del código.

Características:

- Compilación estática.
- Reactividad sin Virtual DOM.
- Integración nativa con TypeScript para validación de tipos.
- Sintaxis minimalista.

Historia y Evolución:

Creado por Rich Harris en 2016, Svelte ganó popularidad por su enfoque innovador frente a React o Vue. TypeScript, desarrollado por Microsoft en 2012, se adoptó ampliamente para mitigar errores en JavaScript.

Ventajas:

- Rendimiento superior.
- Menor sobrecarga de código.
- Facilidad de aprendizaje.

Desventajas:

- Ecosistema más pequeño vs React/Vue.
- Menor adopción empresarial.

Casos de Uso:

- Aplicaciones de una página (SPA).
- Herramientas de visualización de datos en tiempo real.

Casos de Éxito:

- Plataforma de análisis de The New York Times.
- Aplicación de gestión de tareas Todoist.

1.2 Backend: FastAPI

Definición:

Framework Python para construir APIs RESTful y WebSockets con tipado dinámico y soporte asíncrono.

Características:

- Generación automática de documentación (OpenAPI).
- Soporte para async/await.
- Validación de datos con Pydantic.

Historia:

Lanzado en 2018 por Sebastián Ramírez, se popularizó por su velocidad y simplicidad en comparación con Django o Flask.

Ventajas:

- Alto rendimiento (comparado con Node.js).
- Curva de aprendizaje suave.

Desventajas:

- Menor madurez en herramientas de ORM.
- Limitado para aplicaciones monolíticas complejas.

Casos de Uso:

- Microservicios para fintech.
- APIs para IoT.

Casos de Éxito:

- Uber (herramientas internas).
- Netflix (prototipos rápidos).

1.3 Base de Datos: ELK (Elasticsearch, Logstash, Kibana)

Definición:

Stack para análisis y visualización de datos en tiempo real.

Elasticsearch: Motor de búsqueda.

Logstash: Pipeline de procesamiento.

Kibana: Dashboard de visualización.

Historia:

Elasticsearch (2010) surgió como alternativa a Solr, evolucionando hacia un ecosistema completo para big data.

Ventajas:

- Escalabilidad horizontal.
- Consultas en tiempo real.

Desventajas:

- Complejidad de configuración.
- Costo elevado en escalamiento.

Casos de Uso:

- Monitoreo de logs en aplicaciones distribuidas.
- Análisis de comportamiento de usuarios.

Casos de Éxito:

- Netflix (gestión de logs).
- LinkedIn (búsqueda de perfiles).

1.4 Arquitectura Dirigida por Eventos (EDA)

Definición:

Patrón donde componentes se comunican mediante eventos asíncronos, desacoplados de la fuente.

Características:

- Escalabilidad mediante brokers (Ej: Kafka).
- Tolerancia a fallos.
- Respuesta en tiempo real.

Ventajas:

- Flexibilidad para integrar sistemas heterogéneos.

Desventajas:

- Complejidad en trazabilidad de eventos.
- Latencia en procesamiento batch.

Casos de Uso:

- Sistemas de pagos distribuidos.
- Plataformas de IoT.

Ejemplo:

- Uber (coordinación de viajes).

1.5 EMF: Componentes (Com)

Definición:

Modelado de software mediante componentes reutilizables y encapsulados.

Ventajas:

- Reusabilidad y mantenibilidad

Desventajas:

- Diseño inicial complejo.

Casos de Uso:

- Sistemas ERP modulares.
- Librerías UI (Ej: Svelte components).

1.6 EMB: Capas

Definición:

Organización del código en capas (presentación, lógica, datos).

Ventajas:

- Separación de responsabilidades.

Desventajas:

- Acoplamiento vertical rígido.

Casos de Uso:

- Aplicaciones empresariales tradicionales.
- Plataformas educativas.

1.7 Integración: WebSockets

Definición:

Protocolo de comunicación bidireccional en tiempo real.

Ventajas:

- Baja latencia.

Desventajas:

- Complejidad en balanceo de carga.

Casos de Uso:

- Chats (Slack).
- Juegos multijugador.

2. Relaciones entre Temas

Svelte + WebSockets: Actualización en tiempo real de interfaces.

FastAPI + ELK: Logging estructurado de APIs.

EDA + Capas: Eventos como puente entre capas de negocio y datos.

Componentes + Svelte: Desarrollo modular frontend.

2.1 Popularidad del Stack

Tema	Popularidad
Svelte + TS	Creciente (15% uso en 2023 vs React 75%)
FastAPI	Alta en Python (30% adopción en nuevas APIs)
ELK	Dominante en logging (60% mercado)
WebSockets	Estándar en apps real-time (80% adopción)

2.2 SOLID vs Stack

Principio	Svelte+TS	FastAPI	ELK	EDA
Single Resp.	Componentes	Endpoints	Roles separados	Eventos únicos
Open/Closed	Herencia limitada	Extensible	Plugins	Nuevos listeners
Liskov	No aplica	Interfaces Pydantic	Schemas	Contratos eventos
Segregación	Props opcionales	Routers modular	Índices específicos	Canales dedicados

Inversión	Menos común	DI en dependencias	Plugins	Middleware
------------------	-------------	--------------------	---------	------------

2.3 Atributos de Calidad vs Stack

Atributo	Svelte+TS	FastAPI	WebSockets	EDA
Rendimiento	Alto (sin VDOM)	Async I/O	Baja latencia	Escalabilidad
Escalabilidad	Limitada	Horizontal	Conexiones	Alta (brokers)
Mantenibilidad	Componentes	Docs automáticas	Estado persistente	Desacople

2.4 Tácticas vs Stack

Táctica	Svelte+TS	ELK	EDA	WebSockets
Caching	Store local	Cache de índices	Eventos recurrentes	Sesiones
Balanceo	SSR	Shards	Particionado	Proxy (ej. NGINX)

Monitorización	Integra ELK	Kibana	Trazas	Ping/Pong
----------------	-------------	--------	--------	-----------

2.5 Patrones vs Stack

Patrón	Svelte+TS	FastAPI	EDA	Capas
MVC	Componentes	Routers	Consumidores	Capa presentación
Observer	Reactividad	WebHooks	Publicar/Suscribir	No aplica
Singleton	Stores	Conexiones DB	Brokers	Servicios

2.6 Mercado Laboral vs Stack

Tema	Demanda (1-5)	Salario Promedio (USD)	Industrias Clave
Svelte + TS	3	90k– 90k–120k	Startups, Tech
FastAPI	4	100k– 100k–140k	Fintech, Cloud

ELK	5	110k– 110k–150k	IT, E-commerce
WebSockets	4	95k– 95k–130k	Gaming, Comunicaciones

Repositorio GitHub

<https://github.com/SouthThunder/Taller2>

Diapositivas

https://www.canva.com/design/DAGITEM4kO8/o1pzNxCGiCg4DoGSAp_VoQ/edit?utm_content=DAGITEM4kO8&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton