

vantegrate.

CLASE DE LOS JUEVES • COMUNIDAD SALESFORCE

Desarrollar con **IA** en Salesforce

Casos reales y los trucos que nadie te cuenta

Cómo integramos la IA al día a día de proyectos de Salesforce, con demos en vivo sobre una org real. Validación de requerimientos con prototipos, MCP y Headless 360, operaciones de datos en lenguaje natural y creación de agentes con vibe coding. Sin humo.

Demos en vivo

Casos de proyectos reales

Masterclass

SPEAKERS • TDX '26 • SAN FRANCISCO

Esteban Morales

Senior Salesforce Architect en Vantegrate

TDX SPEAKER • DREAMFORCE SPEAKER

Francisco Morales

CEO y Co-founder de Vantegrate • Founder y CEO de Ventix

TDX SPEAKER • DREAMFORCE SPEAKER

Material de la clase publicado por Vantegrate.

Más contenido en vantegrate.com y en el blog vantegrate.com/pulse.

Qué veremos

Seis bloques: tres conceptos y tres demos en vivo sobre una org real, más los aprendizajes de aplicar esto todos los días en proyectos con clientes.

- | | | |
|----------|--|----------------|
| 1 | El cambio de mentalidad
Por qué la diferencia no es la herramienta: es el flujo de trabajo. | CONCEPTO |
| 2 | De un Excel de un cliente a un LWC funcional
Validación de requerimientos con prototipos, antes de escribir código productivo. | DEMO |
| 3 | Qué es un MCP, más Headless 360
El protocolo que conecta a los LLMs con tu org, sin navegador. | CONCEPTO |
| 4 | Datos en lenguaje natural
Consultar, transformar y actualizar la org conversando. Y sus limitaciones reales. | DEMO |
| 5 | Agentforce con vibe coding
Agentscript nativo más el truco de la skill en Cursor para no quemar créditos. | DEMO |
| 6 | Trucos del camino
Lo que aprendimos aplicando esto todos los días: lo bueno y lo malo. | BEST PRACTICES |

1 • CONCEPTO • PUNTO DE PARTIDA

La diferencia no es la herramienta: es el **flujo de trabajo**

La mayoría de los equipos usa la IA para autocompletar código. Los que la usan bien están entregando en días lo que antes tomaba sprints.

IA como autocompletado

- ✗ Se usa recién en la etapa de desarrollo, cuando el requerimiento ya está cerrado.
- ✗ El error de especificación se descubre en la demo del sprint.
- ✗ Cada integración de datos sigue siendo export, Excel y Data Loader.
- ✗ La IA acelera tipeo, no decisiones.

IA como flujo de trabajo

- Entra desde la validación del requerimiento: prototipos que el cliente toca.
- El error de especificación se descubre en la primera reunión.
- Los datos se operan en lenguaje natural contra la org, con revisión.
- La IA ejecuta y amplifica; vos dirigís, juzgás y decidís.

Todo lo que sigue sale de **proyectos reales con clientes**: mismos casos, mismas herramientas, mismos errores que cometimos nosotros primero.

2 • DEMO • VALIDACIÓN DE REQUERIMIENTOS

De un Excel de un cliente a un **LWC funcional**

El requerimiento vive en un Excel. En lugar de arrancar por el desarrollo, lo prototipamos con Claude, lo validamos con el usuario final y recién después escribimos código productivo.

01 · ORIGEN

Excel del cliente

La lógica de negocio real: reglas, columnas, fórmulas. El punto de partida.

02 · PROTOTIPO

Prototipo con Claude

Un artefacto web funcional que vive online, generado en minutos.

03 · VALIDACIÓN

Prueba con usuarios

El usuario final lo prueba, lo rompe y lo corrige antes del desarrollo.

04 · CÓDIGO

LWC con Cursor o Claude Code

El HTML validado es el spec: se convierte en un Lightning Web Component real.

El cambio de mentalidad: **validar antes de desarrollar**. El prototipo cuesta minutos; el retrabajo de un LWC mal especificado cuesta sprints.

El caso: un cotizador que vivía en Excel

Escenario real: el cliente cotiza con una planilla con reglas de negocio embebidas. La convertimos en un prototipo con la UX de Salesforce que vive en un tab de la Opportunity, y el cliente lo valida tocándolo.

Las reglas del Excel

Descuento máximo 10% sin aprobación, margen mínimo 20% por línea, IVA 21%, precios en USD, validez de 15 días.

Reglas editables en pantalla

Los parámetros se ajustan desde la misma UI para probar escenarios con el cliente en la reunión.

Lo que el vendedor no ve

El margen queda oculto en la vista del vendedor, pero la regla se sigue aplicando por detrás. Detalle que salió de la validación.

Ese último detalle es el punto: **salió de que el usuario tocara el prototipo**, no de un documento de requerimientos de 40 páginas que nadie lee.

3 · CONCEPTO · MODEL CONTEXT PROTOCOL

Qué es un MCP

El **Model Context Protocol** es un estándar abierto creado por Anthropic (fines de 2024) que define cómo un LLM descubre y usa herramientas, datos y sistemas externos. Pensalo como el USB-C de la IA: una sola forma de conectar cualquier modelo con cualquier sistema.

API tradicional

- Contrato fijo: endpoints y parámetros definidos de antemano.
- Un desarrollador escribe código custom para cada integración.
- El que decide qué llamar y cuándo es el **programa**.
- Cambia la API y hay que reescribir el cliente.

MCP

- El servidor **describe sus herramientas** y el modelo las descubre en runtime.
- Una integración sirve para cualquier cliente MCP: Claude, Cursor, IDEs.
- El que decide qué herramienta usar y cuándo es el **modelo**, según lo que le pedís en lenguaje natural.
- Nuevas herramientas: el agente las incorpora sin reescribir nada.

Una API expone datos y acciones a **sistemas**. Un MCP expone habilidades y conocimiento a **agentes**.

CONCEPTO • TDX 2026

Headless 360: Salesforce sin navegador

Anunciado en TDX 2026: toda capacidad de la plataforma, datos, workflows y lógica de negocio, queda expuesta como **API, herramienta MCP o comando CLI**. El navegador pasa a ser opcional: humanos y agentes operan Salesforce desde cualquier superficie.

01 • DISCOVER

Búsqueda semántica sobre todas las APIs y skills: el agente encuentra la capacidad que necesita.

02 • DESCRIBE

Devuelve el contrato técnico: parámetros, dependencias y pasos ordenados.

03 • DISPATCH

Ejecuta la skill elegida, respetando los permisos y el trust layer del usuario.

04 • READ-ONLY

La versión de solo lectura de dispatch, para operar con red de seguridad.

Todo pasa por el **mismo modelo de seguridad de tu org**: perfiles, permisos y trust layer. El agente no puede más que el usuario autenticado.

4 • DEMO • OPERACIONES SOBRE DATOS

Operá tu org **conversando**

Conectamos Claude al MCP de Salesforce y operamos sobre la org en lenguaje natural: consultar, transformar, actualizar y mover datos. Sin herramientas de migración, sin CSVs infinitos.

// Lo que antes era: export CSV, Excel, limpiar, Data Loader y rezar

Vos: "Encontrá las Accounts duplicadas, marcá cuál es la principal y completales el Industry que les falta."

Claude: consulta, razona los duplicados, propone el plan, vos aprobás y ejecuta los updates.

Consultas

De pregunta de negocio a SOQL, sin abrir el Developer Console.

Transformaciones

Mapeos, deduplicación y limpieza razonadas por el modelo, con validación previa.

Escritura controlada

Updates con revisión antes de ejecutar: vos aprobás, el agente ejecuta.

Las limitaciones reales, porque sin humo

Esto también lo aprendimos usándolo: el MCP no reemplaza todas las herramientas clásicas todavía. Saber dónde corta es parte de usarlo bien.

Updates de a un registro por vez. Las herramientas del MCP hoy operan sobre el REST API de sObjects: sin Composite ni Bulk API. Para 40 registros va bien; para 40.000, seguí usando Data Loader o Bulk API.

El contexto no es infinito. Traer miles de filas a la conversación no escala: conviene filtrar con SOQL y trabajar por lotes chicos.

El modelo puede equivocarse con confianza. Por eso el flujo es propone, revisás, ejecuta. Nunca al revés en una org productiva.

Los permisos mandan. El agente hereda tu usuario: si no podés borrar un registro desde la UI, el agente tampoco. Eso es una feature, no un bug.

5 • DEMO • AGENTES

Creando Agentforce con **vibe coding**

Construimos un agente con **Agentscript**: describimos el comportamiento en lenguaje natural y el vibe coding nativo de Salesforce genera el agente. Y como no todo es color de rosa, mostramos la alternativa.

Vibe coding nativo (Agentforce Vibes)

- Describís el agente y se genera el Agentscript: topics, instrucciones y acciones.
- Iterás conversando, con el contexto de tu org.
- Integrado al ciclo de vida Agentforce: deploy y testing en la plataforma.
- El costo: consume **créditos** de la plataforma.

El truco: una skill en Cursor

- Una skill propia que enseña a Cursor a escribir Agentscript correcto.
- Mismo resultado: el agente se genera y despliega vía CLI.
- No depende de créditos: usás tu propia suscripción de IA.
- Iterá barato en desarrollo y dejá los créditos para producción.

El aprendizaje real: el vibe coding no reemplaza el criterio. **El que dirige, juzga y decide seguí siendo vos**; el agente ejecuta y amplifica.

6 • BEST PRACTICES • APRENDIZAJES

Los **trucos** que aprendimos en el camino

Validá antes de desarrollar. El prototipo es la especificación: si el cliente no lo tocó, todavía no hay requerimiento.

Dale contexto real, no descripciones. El Excel del cliente, un screenshot de la org, el objeto con sus campos: el modelo trabaja mucho mejor con artefactos que con prosa.

Prompts que compilan. Pedí código para tu org: nombres de API reales, versión, límites. Un "hacé un LWC" genérico genera deuda; "hacé un LWC para este objeto con estos campos" genera valor.

Revisá antes de cada escritura masiva. Que el agente proponga el plan y muestre el resumen antes de tocar un registro. Siempre.

Iterá donde es barato. Desarrollo y pruebas con tu suscripción (Cursor o Claude); los créditos de la plataforma, para lo que va a producción.

El criterio no se delega. La IA multiplica lo que ya sabés de Salesforce; no reemplaza saber a dónde querés llegar.

COMUNIDAD

Sumate a la comunidad de Salesforce en español **más grande del mundo**

Más de 15 países, clases todos los jueves, y un grupo donde se comparten dudas, vacantes y todo lo nuevo del ecosistema. El grupo se llama Masterclass de Salesforce y el acceso está en vantegrate.com.

15+ países

Clases todos los jueves

Grupo Masterclass de Salesforce

E-LEARNING • [VANGUARD.VANTEGRATE.COM](https://vanguard.vantegrate.com)

Qué es **Vanguard**

Nuestra plataforma de e-learning para aprender y certificarte en Salesforce: cada curso sigue el temario oficial de una certificación, de la teoría al examen. Cursos disponibles de Omnistudio, MuleSoft, Platform App Builder, Service Cloud y Agentforce en vanguard.vantegrate.com.

Conceptos teóricos

El temario de cada certificación explicado en español, lección por lección.

Ejercicios prácticos propios

Práctica real en org diseñada por nosotros, no solo teoría para memorizar.

Exámenes mock

Serie de simulacros por certificación para llegar al examen real sin sorpresas.

CHALLENGE • SE GANA EN VIVO

Querés ganarte un **acceso a Vanguard**

Vamos a necesitar lo siguiente, y de paso practicás lo que vimos en la clase:

- 1 Entrá al blog de Pulse**
vantegrate.com/pulse, elegí el artículo que más te interese.
- 2 Pedile a tu LLM amigo que te haga un resumen**
Claude, ChatGPT, Gemini, el que tengas a mano. Ya viste cómo sacarle el jugo.
- 3 Compartí el resumen en el grupo de WhatsApp**
El grupo Masterclass de Salesforce, así también lo lee el resto de la comunidad.
- 4 Listo**
Te va a llegar un acceso a Vanguard.

GRACIAS POR ACOMPAÑARNOS

Nos vemos el próximo jueves

Te llevás el workflow completo, los prompts y la skill de las demos, y un checklist para tu próximo proyecto de Salesforce con IA.

Sitio	vantegrate.com
Blog	vantegrate.com/pulse
E-learning	vanguard.vantegrate.com
Glosario	vantegrate.com/glosario

Vantegrate desarrolla y comercializa agentes de inteligencia artificial para organizaciones medianas y grandes, y sostiene la comunidad de Salesforce en español más grande del mundo con la Clase de los Jueves.