

CURSO DE POSGRADO

Técnicas y Gestión de las Pruebas de Software

Gustavo Guimerans

DOCENTE

Niveles de pruebas

De módulo, componente o unitaria

- Noción de **Prueba “Unitaria”**
- Objetivos
- **Diseño de Casos de Prueba**
- **Construcción del entorno**
- ¿Quién?
- **Gestión y Ejecución**
- Conclusiones

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

- **Noción de Prueba Unitaria**

Pruebas de componentes software individuales.

- **Nivel de Pruebas Unitarias o de Componentes**

Fase de trabajo dentro de un proceso de desarrollo, posterior a la programación de las unidades o componentes del software.

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

Objetivos: *Validación de los componentes de base del sw.*

- *La especificación y diseño.*
- *Adecuación entre especificación/diseño y realización.*
- *Probar que la construcción satisface los requerimientos funcionales en situaciones normales de ejecución.*
- *Detectar tempranamente (y corregir) defectos:*
 - *desde no respeto de nomenclatura, normas, reglamentaciones, ...*
 - *hasta errores algorítmicos, no respeto de la especificación o baja calidad en características no-funcionales...*
- *Construir una base sólida sobre la cuál apoyar el resto de la construcción e integración del software.*

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

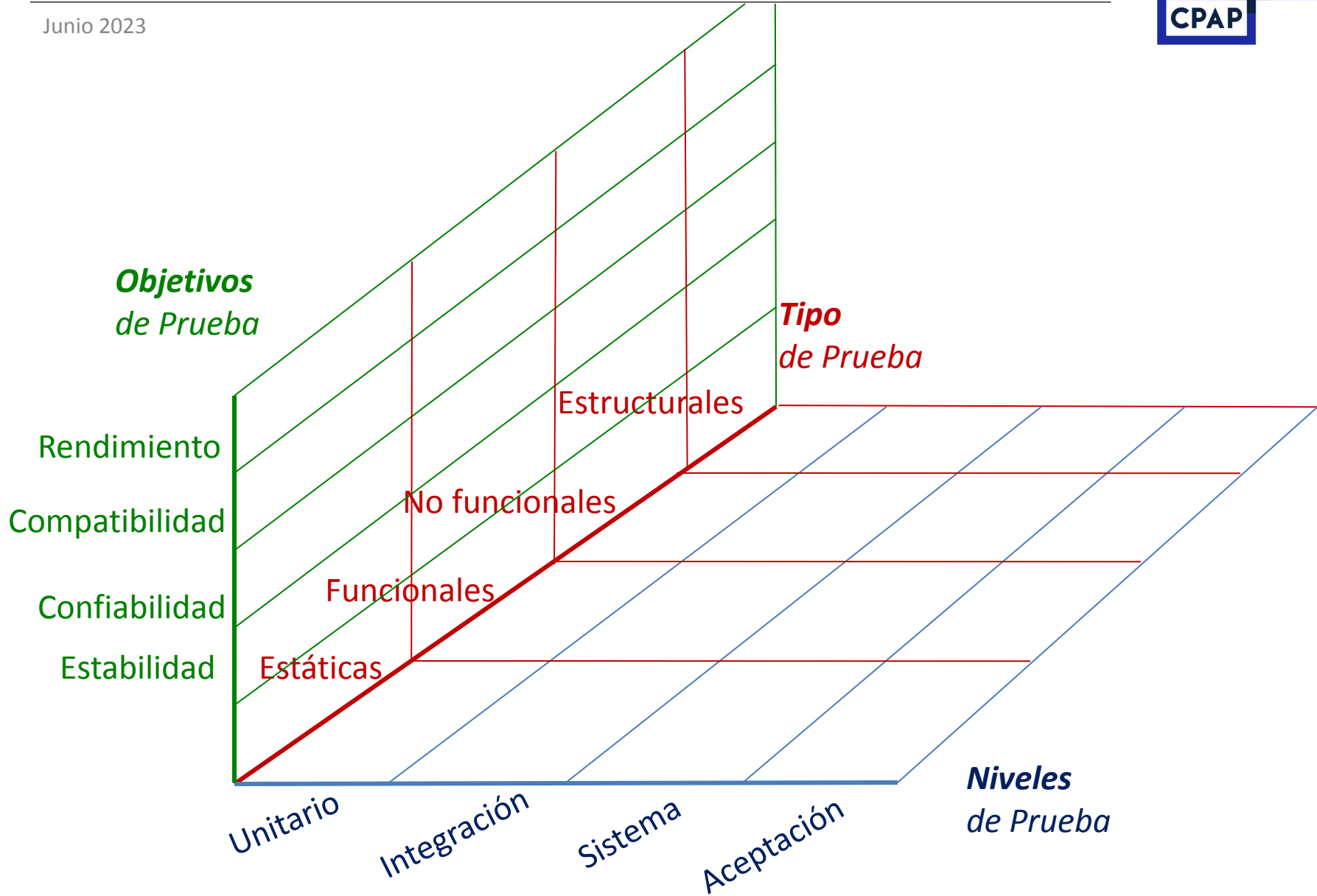
De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

Diseño de los casos de prueba.

- *Identificación de los objetivos*
- *Análisis de riesgo del componente*
- *Selección de objetivos a ser probados*
- *Plan de pruebas unitarias*
- *Construir un plan progresivo, de acuerdo a la estrategia general guiada por objetivos*
- *Cobertura, aspirar a...*
 - *100% cobertura de decisiones y*
 - *100% de cobertura funcional por Particiones de Equivalencia*
- *Gestión de incidentes*



De módulo, componente o unitaria

Plan de pruebas.

Analizar para cada característica qué método utilizar:

- *Revisión de la especificación, normas, incidentes*
- *Cobertura de Decisión o de Condición*
- *Particiones de Equivalencia y Valores Límites*
- *Atributos de calidad*

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

Construcción del entorno.

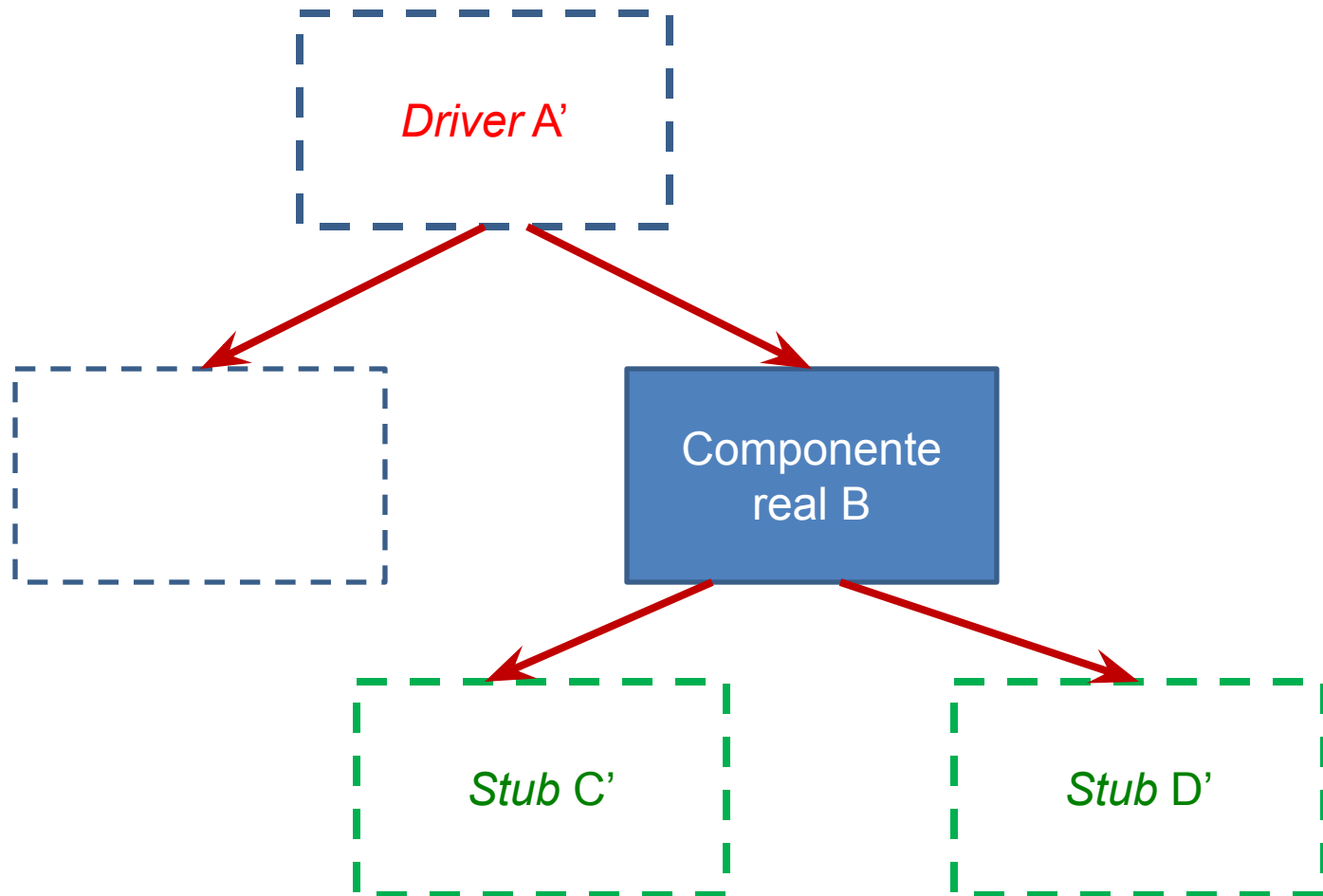
- *Marco de trabajo de pruebas unitarias*
 - *Herramienta que aporta un entorno para realizar pruebas unitarias o de componentes en el cual el componente puede ser probado de forma aislada o con los componentes de sustitución (“stubs”) o controladores (“drivers”) adecuados. También presenta otros soportes para el desarrollador tales como capacidades para depurar.*
- *“Arnés” de pruebas unitarias*
 - *Entorno de pruebas constituido por stubs y drivers necesarios para llevar a cabo la prueba.*

De módulo, componente o unitaria

Construcción del entorno.

- *Stub (Componente de Sustitución)*
 - *Un esqueleto o implementación para un propósito especial de un componente software, usado para desarrollar o probar un componente al que se llama o es de otra forma dependiente en él. El reemplaza un componente llamado.*
- *Driver (Controlador de prueba)*
 - *Un componente de software o herramienta de pruebas que sustituye a un componente y que asume el control y/o las llamadas a un componente o un sistema.*

De módulo, componente o unitaria



De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

¿Quién?

- Depende del proceso y el equipo.
- Normalmente el equipo de desarrollo.
- El “desarrollador”
 - en general la misma persona que lo implementó;
 - es positivo el conocimiento detallado del módulo a probar;

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- ¿Quién?
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

Gestión y Ejecución.

- *Plan de Pruebas, Especificaciones de Casos de Prueba, Reporte de pruebas ejecutadas, Reporte de casos de incidentes.*
- *Posible utilización del marco normativo.*
- *Aspirar a utilizar procedimientos formalizados y apoyados con herramientas (desarrollo/prueba/corrección)*

De módulo, componente o unitaria

Gestión y Ejecución. Seguimiento.

- *Centralizar estado de avance y resultados*
 - *Poder comparar el avance contra lo planificado*
 - *Poder identificar cada caso de prueba y conocer su estado*
 - *Poder hacer síntesis para evaluación de:*
 - *Cobertura (para cada técnica elegida)*
 - *Medidas de fiabilidad (al menos cantidad de errores por tipo)*
- *Prioridad a la estimación del avance del proyecto...*
 - *Poder brindar información precisa al Jefe de Proyecto y a clientes*
- *Pero prever medidas para el análisis de mejoras (posmortem)*
 - *La información recabada durante las pruebas es fundamental para identificar problemas sistemáticos y proponer mejoras para los proyectos futuros*

De módulo, componente o unitaria

Gestión y Ejecución. Enfoque TDD y xUnit

- *Ej. Herramientas JUnit y NUnit*
- *Casos de pruebas de componentes incluso antes de tener el código disponible (“haber programado”)*
- *Casos de prueba integrados al ambiente de desarrollo*
- *Monitor de ejecución de pruebas*
- *(re-)ejecución de pruebas “cada vez” que se modifica el código*
- *Enfoque integrado a herramientas a distintos IDEs*

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

- Noción de Prueba Unitaria
- Objetivos
- Diseño de Casos de Prueba
- Construcción del entorno
- Gestión y Ejecución
- Conclusiones

De módulo, componente o unitaria

Conclusiones

- *Detección temprana de defectos - costos.*
- *Construcción de software ascendente, usando componentes robustos (sólidos y probados).*
- *Reduce problemas de integración y sistema.*
- *Orientado a funcionalidades y robustez.*
- *Objetivo de “100% de cobertura”*
 - *Cobertura de Decisión y cobertura de Particiones de Equivalencia;*
 - *Valores Límites;*
- *Incluir “todos los casos” de detección de errores.*
- *Implementar “Arnés de prueba”.*
- *Permite de acelerar/automatizar la ejecución de casos de prueba.*

De módulo, componente o unitaria

Función: Autenticación.

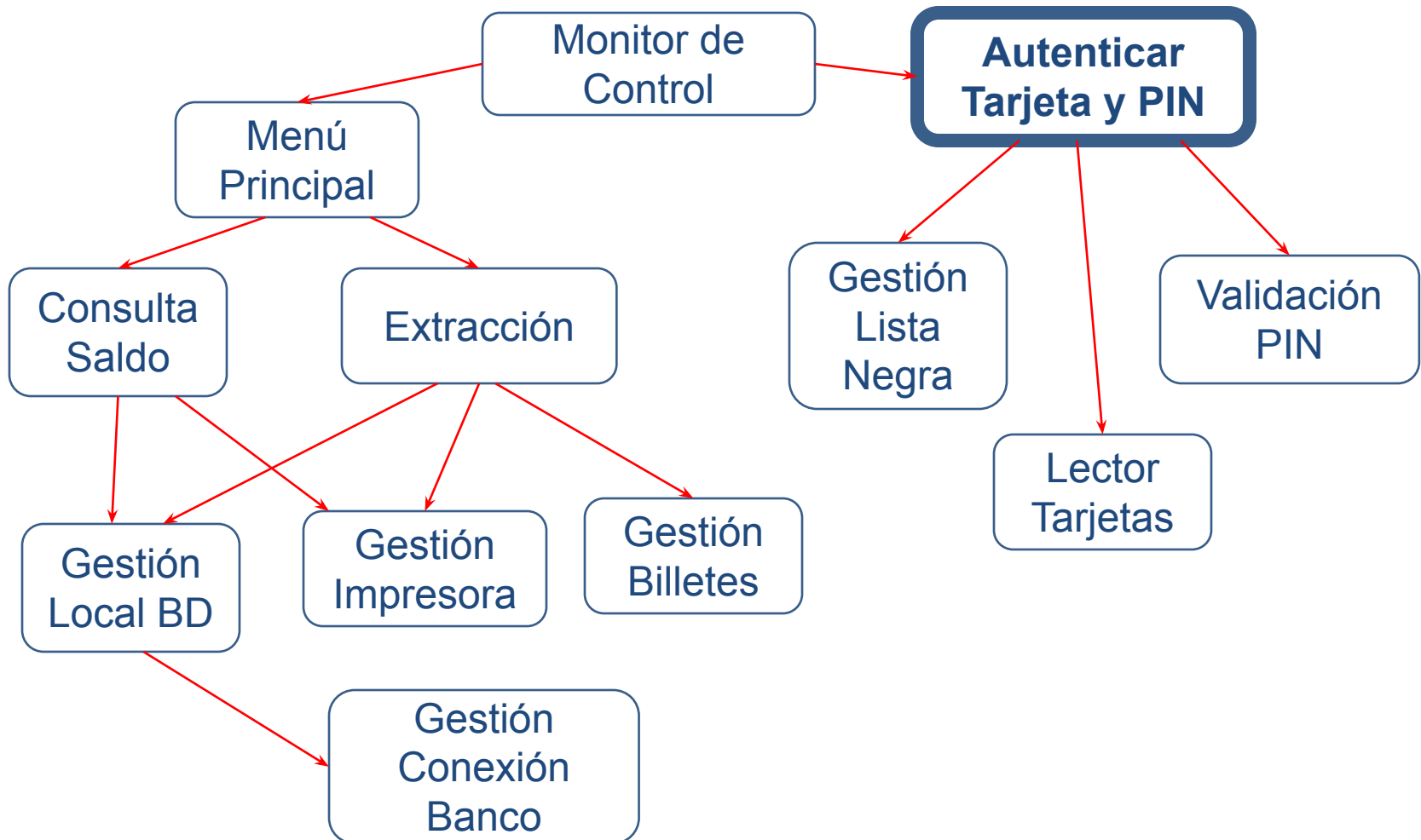
- *El usuario ingresa su tarjeta bancaria y espera para ingresar su código. El lector de tarjetas extrae el número de tarjeta. La función sólo aceptará tarjetas que no se encuentren en la “lista negra”. Si la tarjeta se encuentra en la “lista negra” será “tragada” por el cajero automático. El usuario dispone de hasta una cantidad “MAX” de intentos de ingreso de su número de identificación. En caso de superar esa cantidad “MAX” la autorización será denegada y la tarjeta será “tragada”. Si la tarjeta no está en la “lista negra” y si el usuario ingresa correctamente su número de identificación la función autorizará al usuario. Se utilizarán las variables de salida “estatus_tarjeta”, “estatus_número_identificacion”, “autorizacion”, y “numero_de_tarjeta”.*
- *El pseudo-código y los fuentes están disponibles*

De módulo, componente o unitaria

Función: Autenticación.

- *Proponer Plan de Pruebas Unitarias*
 - *Objetivos perseguidos*
 - *Entorno*
 - *Estrategia*
 - *Condiciones de salida*

De módulo, componente o unitaria



Puesta en común

¿Pausa?

Junio 2023

Prueba de Integración

- Noción de **Prueba de Integración** y Objetivos
- **Estrategias de Integración**
- **Concepción de casos de prueba**
- ¿Quién?
- Conclusiones

Junio 2023

Prueba de Integración

- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Prueba de Integración

- *Pruebas diseñadas y ejecutadas con el objetivo de encontrar **defectos** en las **interfaces** e **interacciones** entre **elementos integrados**.*
- *Cada paso de la integración agrupa **dos o más componentes** en un único grupo integrado, el que es probado como una unidad por sí sola*
- *Todos los tipos de pruebas son posibles*
 - *Revisión estática*
 - *Estructurales a través de Grafos de Llamada entre componentes*
 - *Funcionales, observando las funciones o servicios proporcionados por el primer nivel de acceso del grupo integrado*
 - *No-funcionales, observando ese mismo grupo integrado*
 - *...*

Junio 2023

Prueba de Integración

- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Junio 2023

Prueba de Integración

- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Prueba de Integración

Estrategias de Integración

- *Establecer la secuencia en la que los componentes serán integrados*

- *Cinco estrategias más comunes*
 - *Big Bang*
 - *Descendente*
 - *Ascendente*
 - *Vertical*
 - *Flujo de Datos*

Prueba de Integración

Estrategias de Integración. Big Bang.

- *Tipo de pruebas de integración en el que los elementos software, elementos hardware o ambos son combinados en un componente o un sistema global en lugar de hacerlo por fases.*
- *Ventajas*
 - *No requiere un plan incremental*
 - *No requiere construir Stub ni Drivers*
- *Inconvenientes*
 - *No permite generar pruebas de “interface” entre niveles inferiores*
 - *No permite generar anticipadamente pruebas de robustez*
- *Es una forma de saltar el nivel de integración y hacer directamente pruebas de nivel Sistema.*

Prueba de Integración

Estrategias de Integración. Descendente.

- *Enfoque incremental de las pruebas de integración donde el componente en el nivel más alto en la jerarquía es probado primero, con los componentes del nivel inferior simulados mediante stubs. Los componentes probados son usados luego para probar a los componentes del nivel inferior. El proceso se repite hasta que el nivel más bajo haya sido probado.*
- **Ventajas**
 - *Aspectos funcionales y lógica del negocio son probados prioritariamente.*
 - *Orientado a validar funcionalidades.*
- **Inconvenientes**
 - *La calidad se descubre al final*

Prueba de Integración

Estrategias de Integración. Ascendente.

- *Enfoque incremental de pruebas de integración donde los componentes de más bajo nivel son probados en primer lugar, posteriormente son utilizados para facilitar las pruebas de componentes de un nivel superior. Este proceso se repite hasta que es probado el componente en el extremo superior de la jerarquía.*
- *Ventajas*
 - *No requiere Stubs, solo Drivers.*
 - *Excelente para probar robustez y aspectos técnicos y de construcción del software en general*
- *Inconvenientes*
 - *Problemas funcionales o de lógica del negocios son validados al final*

Prueba de Integración

Estrategias de Integración. Vertical.

- *Un enfoque incremental de pruebas de integración donde son agrupados todos los componentes necesarios para **ejecutar completamente una operación o función del nivel superior**. Una vez probados se pasa a otra operación o función del nivel superior, hasta probarlas todas.*
- *Típicamente se trata de agrupar las operaciones o funciones visibles a nivel del usuario final, del menú, o de servicios.*
- *Ventajas*
 - *Requiere un mínimo de Stubs.*
- *Inconvenientes*
 - *Similares al Big Bang. Anticipación.*

Prueba de Integración

Estrategias de Integración. Flujo de Datos.

- **Flujo de Datos**

Representación abstracta de la secuencia y posibles cambios de estado de los objetos de datos, donde el estado de un objeto puede ser: creación, uso o destrucción.

- **Integración por Flujo de Datos**

Un enfoque incremental de pruebas de integración donde los componentes que forman parte de un mismo flujo de datos son agrupados y probados en conjunto.

- **Ventajas**

- *Adaptado para analizar la dinámica de operaciones, WF, BPM y otros procesos de negocios o empresariales.*

- **Inconvenientes**

- *Es de alto nivel, menor potencial de detección temprana de errores.*

Junio 2023

Prueba de Integración

- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Junio 2023

Prueba de Integración

- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Prueba de Integración

Casos de prueba

- *Combinar ordenadamente los tipos y objetivos*
 - *Estáticas, Estructurales, Funcionales, No-funcionales*
 - *Estabilidad, Confiabilidad, Rendimiento*
- *Utilizar análisis de riesgos para determinar qué casos son los más pertinentes*

- *Estructurales*
 - *Construir grafo donde cada componente es un nodo, y las invocaciones entre componentes son los arcos*
 - *“100% cobertura” de decisiones sobre ese grafo*
 - *Bien adaptadas para evaluar robustez*
 - *Puede usarse para identificar potenciales problemas de rendimiento en los caminos más largos o inusuales*

Prueba de Integración

Casos de prueba

- *Funcionales*
 - *Identificar las funciones de alto nivel del grupo integrado*
 - *No siempre existe una función explícita de ese tipo*
 - *No siempre está especificada*
 - *Concentrarse en la gestión de interfaces, comunicaciones y gestión de incidentes*
 - *Aplicar técnicas de particiones de equivalencia, valores límites, tablas de decisión, casos de uso...*

Junio 2023

Prueba de Integración

- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Junio 2023

Prueba de Integración

- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Prueba de Integración

¿Quién?

- Depende del proceso y el equipo
- Normalmente las realiza el equipo de desarrollo
 - Es necesario el conocimiento de las interfaces y funciones en general

Junio 2023

Prueba de Integración

- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Junio 2023

Prueba de Integración

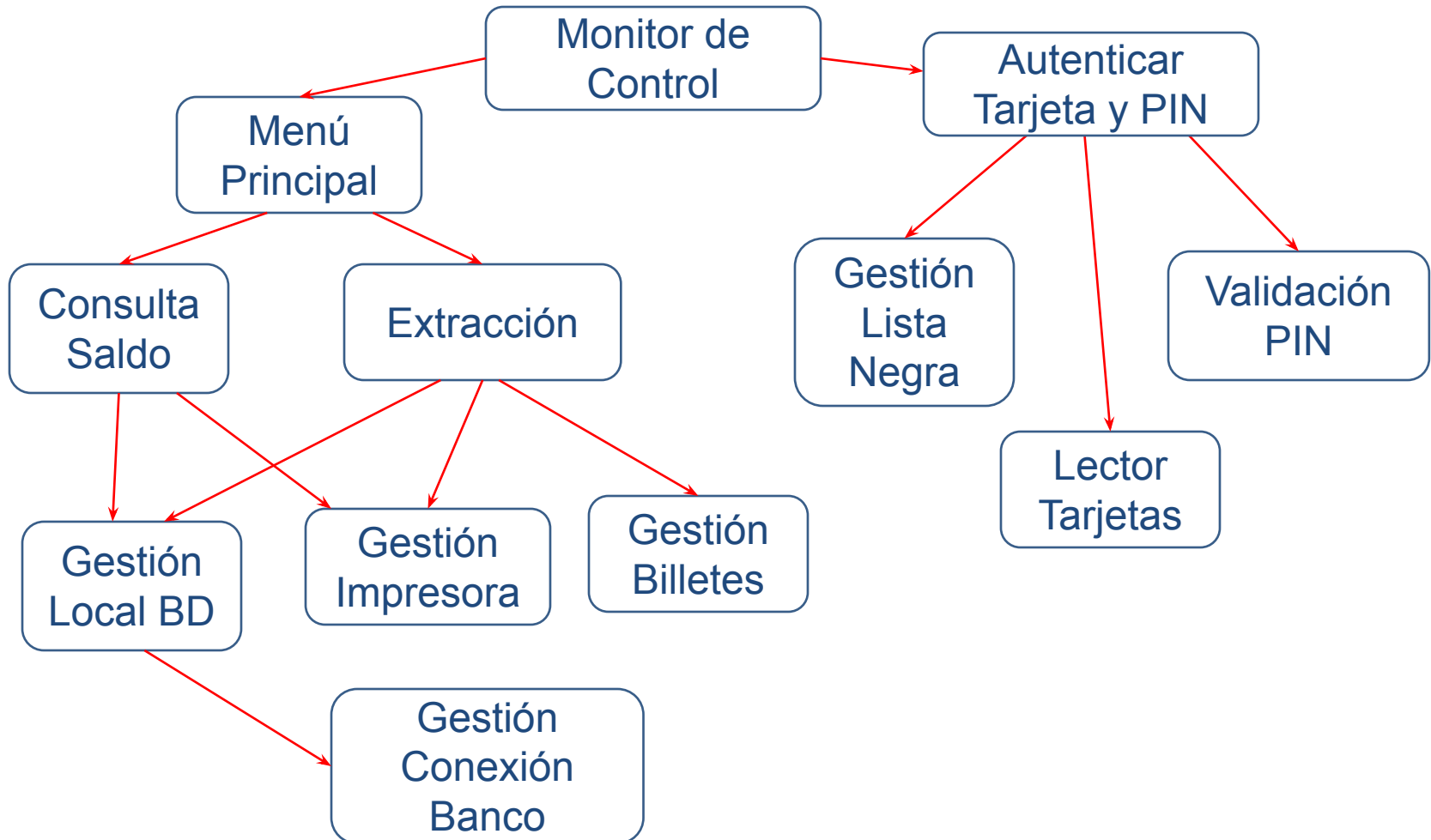
- Noción de Prueba de Integración y Objetivos
- Estrategias de Integración
- Concepción de casos de prueba
- ¿Quién?
- Conclusiones

Prueba de Integración

Conclusiones

- *El nivel de Pruebas de Integración es altamente eficaz para la validación del software*
 - *Permite la detección temprana de problemas de interface entre componentes*
 - *Permite focalizarse en problemas de robustez*
- *Se debe elegir una estrategia de integración adaptada al contexto*
 - *Análisis de riesgos y del contexto*
 - *Coordinar plan de integración con el plan de desarrollo*

Prueba de Integración



Puesta en común

¿Pausa?

Junio 2023

Prueba de Sistema

- Noción de **Prueba de Sistema** y Objetivos
- **Concepción de casos de prueba**
- **¿Quién?**
- **Gestión**
- **Conclusiones**

Prueba de Sistema

- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- Gestión
- Conclusiones

Prueba de Sistema

Noción de Prueba de Sistema y Objetivos

- *El proceso de probar un sistema integrado para verificar que cumple los requisitos especificados.*
- *Adoptar el punto de vista del usuario final*
- *Responsabilidad del “constructor del software”*

- *Misión de las pruebas de sistema*
 - *Obtener información confiable y suficiente acerca del grado en que el sistema ha alcanzado los objetivos especificados*
 - *Tomar una decisión respecto a la entrega del sistema*
 - *Identificar los riesgos potenciales remanentes*

Junio 2023

Prueba de Sistema

- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- Gestión
- Conclusiones

Junio 2023

Prueba de Sistema

- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- Gestión
- Conclusiones

Prueba de Sistema

Concepción de casos de prueba

- *Crear un entorno lo más similar posible al entorno de producción*
 - *Hardware*
 - *Lo más similar posible al previsto para producción, incluyendo características del procesador, memoria RAM, capacidad y velocidad disco, ...*
 - *Redes*
 - *Protocolos, configuración, routers, firewall, ...*
 - *Software de base*
 - *SO, versión y configuración del uso de recursos, ...*
 - *Base de datos*
- *Aplicar técnicas de pruebas de caja negra*
 - *Funcionales*
 - *No-funcionales*

Prueba de Sistema

Concepción de casos de prueba

- *Fuertemente orientados a la conformidad con las especificaciones*
 - *Prioridad a las pruebas de objetivos “nominales” y “estabilidad”*
 - *Identificar principales características no funcionales luego de un análisis de riesgo que determine cuáles son más críticas*
- *Adoptar una visión de conjunto del sistema*
 - *Identificar el contexto real para simularlo, incluir manuales de usuarios, configuraciones, variantes usuales*
 - *Identificar y si es posible, utilizar, algunos usuarios reales durante las pruebas*

Junio 2023

Prueba de Sistema

- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- Gestión
- Conclusiones

Junio 2023

Prueba de Sistema

- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- Gestión
- Conclusiones

Prueba de Sistema

¿Quién?

- *El plan de pruebas de sistema, su diseño y ejecución representan un trabajo importante*
 - *Se requiere de un equipo dedicado y especializado*
 - *Altamente aconsejable que no sea el propio grupo de desarrollo*
- *En general equipo especializado (testers/verificadores)*
- *Equipo de Pruebas*
 - *Equipo de pruebas especializado y con fuerte grado de independencia en los resultados*
 - *Evitar el síndrome de “probar para demostrar que funciona” al que el equipo de desarrollo tiene tendencia.*
 - *Equipo que debe preparar el entorno, identificar las prioridades de prueba y dialogar directamente con el cliente/usuario*
 - *Tareas que pueden resultar difícil de poner en el foco de la actividad del área de desarrollo, en particular en las fases finales de un proyecto*
- *Necesario conocer requerimientos y tener una visión global*

Prueba de Sistema

- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- Gestión
- Conclusiones

Junio 2023

Prueba de Sistema

- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- **Gestión**
- Conclusiones

Prueba de Sistema

Gestión

- *Construir un plan de pruebas*
 - *Objetivos, plan de acción, responsables, condiciones de salida.*
- *Información de gestión*
 - *Avance del trabajo, midiendo la cobertura*
 - *Avance de los resultados, midiendo la fiabilidad*
 - *Herramientas y de procedimientos sistemáticos de gestión*
 - *Registro de los casos de prueba ejecutados y de los resultados*
- *Conocer las condiciones de salida*
 - *Evaluar al ritmo que se avanza hacia su cumplimiento*
 - *Evaluar el riesgo residual*
 - *Ofrecer información de conjunto y objetiva para tomar la decisión de entregar el producto.*

Prueba de Sistema

Gestión. *Desarrollo de software interno a la empresa.*

- *Evitar tendencia a mezclar todos los niveles de prueba*
- *Separar la responsabilidad entre equipo de prueba y usuarios*
 - *El equipo de prueba interno apoya el desarrollo, diseña y ejecuta pruebas durante todos los niveles*
 - *Los usuarios participan en las pruebas de Aceptación*
- *Involucrar a los usuarios una vez que el software esté maduro*
 - *Evitar su participación en etapas aún no probadas*
 - *Si hay demasiados errores se generará una muy mala impresión, difícil de revertir una vez que los defectos hayan sido corregidos.*

Prueba de Sistema

Gestión. *Desarrollo por empresa externa.*

- *Protegerse de los errores mediante cláusulas contractuales*
- *Exigir planes de prueba y reporte de pruebas diseñadas y ejecutadas, y reporte de errores*
- *Exigir un buen nivel de calidad al momento de la entrega, no “meses” después...*

Junio 2023

Prueba de Sistema

- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- **Gestión**
- Conclusiones

Junio 2023

Prueba de Sistema

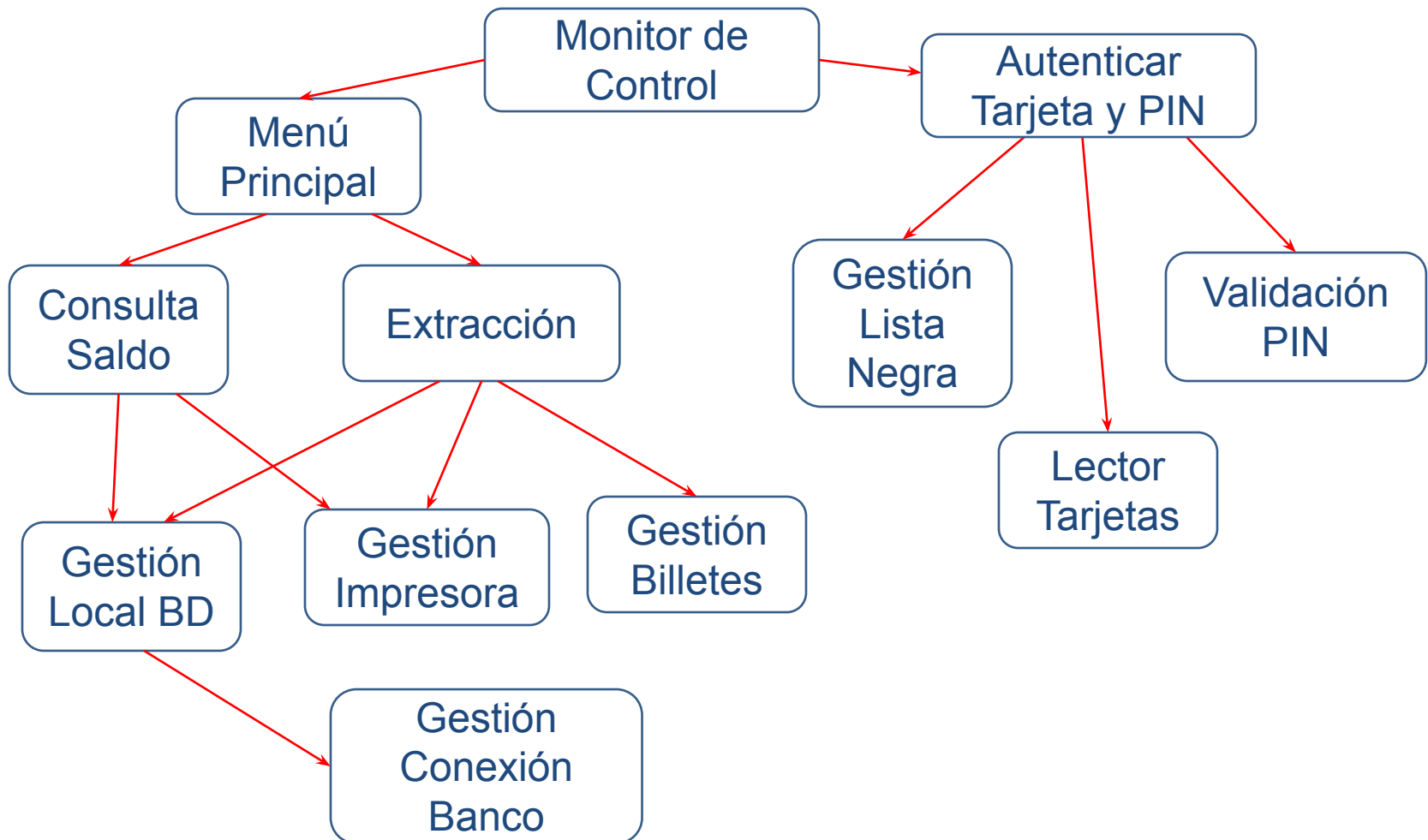
- Noción de Prueba de Sistema y Objetivos
- Concepción de casos de prueba
- ¿Quién?
- Gestión
- Conclusiones

Prueba de Sistema

Conclusiones

- *El nivel de Pruebas de Sistema es necesario para la validación del software*
 - *Permite la detección de problemas de todo tipo*
 - *Se puede contar con un equipo interno o externo, importancia de la especialización, los distintos tipos de pruebas y habilidades*
 - *Si hago hago y si compro exijo a tiempo*
- *El proceso de probar un sistema integrado para verificar que cumple los requisitos especificados.*
- *Adoptar el punto de vista del usuario final*

Prueba de Sistema



Puesta en común

¿Fin?