

Aplicaciones Wisej.NET híbridas, PWA, sin conexión y adaptadas al dispositivo

Guía de la lección · Guía del laboratorio / examen



Guía de la lección

Modos conectado y sin conexión, datos locales, APIs del dispositivo, patrones de conflictos de sincronización, escenarios de campo. Lo que esto significa para un equipo avanzado de Wisej.NET — y cómo leer este módulo.



Enfoque de producción

Modos conectado y sin conexión, datos locales, API de dispositivo, patrones de conflictos de sincronización, escenarios de campo.

1. El problema en producción

Algunos flujos de trabajo empresariales requieren funcionamiento en móvil, tableta, trabajo de campo, quiosco o sin conexión. Los desarrolladores avanzados de Wisej.NET deben decidir cuándo encajan los patrones Hybrid/PWA y cómo conservar un único modelo de aplicación sin ignorar las realidades de los dispositivos.

2. Conceptos clave

Concepto	Por qué importa
Wisej.NET Hybrid	El soporte del framework para ejecutar la misma aplicación en el navegador, como aplicación de escritorio o móvil, y con acceso nativo al dispositivo.
modo conectado	El modelo estándar, en el que la aplicación se ejecuta en el servidor y el dispositivo es un cliente. Es el más sencillo, pero requiere conexión de red.
aplicación local/sin conexión	Ejecutar la aplicación localmente en el dispositivo para que siga funcionando sin conexión, a costa de gestionar los datos locales y la sincronización.
abstracción del dispositivo	Situar la cámara, el sistema de archivos, el GPS y otras funciones del dispositivo detrás de interfaces de servicio para que las pantallas no dependan de una plataforma concreta.
SQLite local	Una base de datos incrustada y ligera en el dispositivo que almacena los datos de trabajo sin conexión y alimenta la cola de sincronización.
cola de sincronización	Una lista duradera de cambios locales que esperan a enviarse al servidor cuando vuelve la conectividad, procesados en orden y reintentados de forma segura.
resolución de conflictos	Las reglas sobre qué ocurre cuando el mismo registro ha cambiado a la vez en local y en el servidor durante la desconexión, decididas por el producto y no por azar.
shell PWA	El manifiesto de la aplicación web progresiva y el service worker que permiten instalar la aplicación en un dispositivo y cargar su shell sin red.

3. Desarrollo de la lección

Trabajar sin conexión es una decisión de producto

El soporte sin conexión no es una casilla que se marca. Cambia la identidad, la vigencia de las autorizaciones, el almacenamiento de datos, la gestión de conflictos, el momento de la auditoría, los procedimientos de soporte y el despliegue. Los desarrolladores avanzados limitan el alcance sin conexión a los flujos de trabajo que realmente lo necesitan.

Mantén el código del dispositivo detrás de servicios

La cámara, el escáner, el portapapeles, las notificaciones y el estado de la red deberían exponerse a través de servicios. El resto de la aplicación debería llamar a interfaces expresadas en términos de negocio en lugar de dispersar código específico del dispositivo por las pantallas.

Cuatro formas de aplicación

Wisej.NET Hybrid permite que un mismo modelo de aplicación se ejecute de cuatro formas. En modo conectado, la aplicación se ejecuta en un servidor y el navegador, o un shell nativo alrededor de un navegador, es un cliente ligero; el dispositivo necesita conexión de red para cada acción. En modo remoto, un shell nativo en el dispositivo se conecta a esa misma aplicación de servidor, pero obtiene acceso a las funciones del dispositivo a través del shell. En modo local, la propia aplicación se empaqueta con el

dispositivo y ejecuta su proceso de servidor en él, así que funciona sin red. El modo sin conexión es el modo local con datos locales y un límite de sincronización con el sistema central.

Cada paso a lo largo de esa línea cambia sencillez por independencia. El modo conectado tiene una sola copia de los datos y un solo lugar donde aplicar las reglas. El modo sin conexión tiene una copia en cada dispositivo, y las reglas deben cumplirse en cada uno de ellos. La forma adecuada es la menos independiente que satisfaga el flujo de trabajo. Un operador de despacho en su mesa sigue conectado. Un técnico en un sótano sin cobertura necesita una cola sin conexión para el trabajo que completa allí.

Indica la forma en la nota de arquitectura antes de diseñar las pantallas, porque decide dónde viven la identidad, los datos y los permisos.

Flujos de trabajo sin conexión y límites de sincronización

Un flujo de trabajo capaz de funcionar sin conexión se diseña en torno a lo que el dispositivo puede guardar y lo que puede cambiar. El modo Field Technician del laboratorio almacena en caché las órdenes de trabajo asignadas al técnico en un almacén SQLite local, y solo esas. Completar una orden de trabajo sin conexión no actualiza el registro; añade un `OfflineCommand` a una cola local con un `SyncState` explícito de pendiente. La pantalla muestra el estado pendiente para que el técnico sepa que la finalización aún no ha llegado a la oficina.

El límite de sincronización es el punto en el que la cola se reproduce contra el servidor. Cada comando se envía en orden, lo ejecuta el mismo servicio de servidor que gestiona el caso en línea y se marca como sincronizado o en conflicto. Como el servicio de servidor es el mismo, cada regla, comprobación de permisos y entrada de auditoría se aplica al trabajo sin conexión exactamente igual que al trabajo en línea. El dispositivo nunca se comunica directamente con la base de datos.

Limita el alcance sin contemplaciones. Cada entidad adicional en caché sin conexión son más datos en un dispositivo que se puede perder, más casos de sincronización que probar y más conflictos que resolver.

Las capacidades del dispositivo detrás de interfaces de servicio

La cámara, el lector de códigos de barras, el GPS, las notificaciones locales y el estado de la red llegan a la aplicación a través del shell Hybrid, y cada uno se comporta de forma distinta en un teléfono, una tableta, un quiosco y un navegador de escritorio. El laboratorio los oculta todos detrás de una abstracción `IDeviceServices`. Una pantalla llama a un método como capturar foto o leer código de barras, y la implementación para la forma actual decide si eso significa la cámara nativa, un selector de archivos o un mensaje de que la función no está disponible.

La abstracción compensa por partida triple. Las pantallas son idénticas en todas las formas, así que un mismo formulario sirve para la oficina y para el campo. Las pruebas pueden sustituirla por un servicio de dispositivo falso que devuelve una foto o un código de barras conocidos. Y un nuevo dispositivo o shell solo requiere una nueva implementación, no buscar llamadas específicas de la plataforma en todos los formularios.

Trata la salida del dispositivo como una entrada no confiable. Un código de barras escaneado se valida en el servidor como cualquier otro valor, y una posición GPS es una pista para el usuario, nunca la prueba de dónde se hizo el trabajo.

Conflictos, permisos obsoletos y seguridad sin conexión

Un conflicto se produce cuando el mismo registro ha cambiado en el dispositivo y en el servidor mientras estaban separados. En la ruta de fallo del laboratorio, el operador de despacho cancela una orden de trabajo mientras el técnico, sin conexión, la completa. Ninguna de las dos versiones es simplemente incorrecta, así que la pantalla de conflictos de sincronización muestra ambas, registra ambas y pide a una persona que decida. Las reglas de resolución, como qué roles pueden imponerse y si una orden cancelada todavía se puede completar, son decisiones de producto documentadas antes de escribir el código.

Los permisos quedan obsoletos sin conexión. Es posible que se haya retirado el rol del técnico mientras estaba desconectado, y el dispositivo sigue guardando el conjunto de permisos de ayer. El límite de sincronización lo gestiona volviendo a evaluar en el servidor cada comando en cola con los permisos actuales, de modo que los comandos en cola de un usuario revocado se rechazan y se auditan en lugar de aplicarse. El dispositivo también refresca sus permisos en caché en cada reconexión.

- Los datos locales se cifran en reposo y se borran al cerrar sesión o tras un periodo de bloqueo.
- Las credenciales en caché caducan, y a partir de entonces un dispositivo robado no puede volver a reproducir comandos.
- Cada comando sin conexión se audita en el servidor con la marca de tiempo original del dispositivo y su hora de sincronización.



Regla avanzada de producción

No te detengas cuando la pantalla parezca correcta. Explica a quién pertenece el estado, las implicaciones de seguridad, las rutas de fallo y el comportamiento en producción antes de dar el módulo por terminado.



Objetivos de aprendizaje

Al terminar este módulo serás capaz de:

- ✓ Explicar las formas de aplicación Hybrid conectada, remota, local y sin conexión.
- ✓ Diseñar flujos de trabajo capaces de funcionar sin conexión con almacenamiento local y límites de sincronización.
- ✓ Usar las capacidades del dispositivo a través de abstracciones de servicio explícitas.
- ✓ Gestionar los conflictos de sincronización, los permisos obsoletos y las contrapartidas de seguridad del trabajo sin conexión.

Guía del laboratorio / examen

Lo que construirás en el laboratorio práctico, el enfoque sugerido y los entregables requeridos.



Laboratorio práctico

Diseña y prototipa un modo Field Technician para EnterpriseOps. Añade un diseño adaptado al dispositivo, una caché de órdenes de trabajo sin conexión, una cola local de finalizaciones, una simulación de sincronización al reconectar y la visualización de conflictos.

Enfoque sugerido

- ✓ Lee el problema en producción e identifica el riesgo de arquitectura antes de tocar la UI.
- ✓ Crea o actualiza primero el límite correspondiente de servicio, comando, modelo o componente.
- ✓ Construye la pantalla o el componente Wisej.NET siguiendo los estándares de proyecto establecidos.
- ✓ Añade una demostración de la ruta de fallo, no solo del caso ideal.
- ✓ Anota lo que ha cambiado en el diario de aprendizaje y actualiza la lista de comprobación correspondiente.

Entregables requeridos

- ✓ Nota de arquitectura Hybrid/sin conexión
- ✓ Abstracción del servicio de dispositivo
- ✓ Modelo de la cola sin conexión
- ✓ Pantalla de conflictos de sincronización
- ✓ Lista de comprobación de usabilidad del modo de campo

Preguntas de repaso

- ✓ ¿Qué datos se permiten sin conexión?
- ✓ ¿Cómo se refrescan los permisos?
- ✓ ¿Qué ocurre cuando los cambios locales entran en conflicto con los cambios del servidor?



Regla avanzada de producción

No te detengas cuando la pantalla parezca correcta. Explica a quién pertenece el estado, las implicaciones de seguridad, las rutas de fallo y el comportamiento en producción antes de dar el módulo por terminado.