

Conceptos previos

- Servidor de aplicaciones
- Multicapa
- json
- REST

Creación del proyecto servidor

- DatasnapREST server wizard
- Escoger FM MainForm (facilita tener Log)
- Escoger “simplified dispatcher”
- Vistazo a los ficheros generados
- Configurar la salida del compilador
- Reorganización de los ficheros
- Testeo con Postman

La Base de Datos

- Scripts para crear la BD: migrations
- Scripts para alimentar la BD: DBSeeds
- FireDac Explorer. Conexión a datos con FireDac

Servicios con DataSnap

- Módulo de OrderService: {METHODINFO ON}
- Registrar la factoria: TDSerClass
- Devolviendo JSON: Server.JSONUtils.pas. Ventajas
- Apuntes sobre Autenticación y Roles

Control de la rutas

- El componente TDSRESTWebDispatcher
- Propiedades DSContext y RESTContext
- Eventos OnParseRequest y OnParsingRequest
- Convenios REST y mapeo de rutas

Notas técnicas sobre la arquitectura

- Mejor extender a MVC + Patrones (Repository, UseCase, UoW)
- Atención a la coordinación de hilos
- Aprovechar multitarea: `unit System.Threading;`
- Tener buen Log

Creacion del proyecto cliente

- Escoger plantilla Tabbed with Navigation
- Vistazo a los ficheros generados
- Configurar la salida del compilador
- Reorganización de los ficheros

Interfaz de usuario con FireMonkey

- FireMonkey: TL;DR
- Creación de la interfaz
- Presentacion: Factor de forma y Estilos
- Importancia de UX profesional

Completando la UX: LiveBinding

- Conceptos básicos de LB
- El componente TPrototypeBindSource

Obteniendo datos del servidor REST

- Server Proxies
- Obteniendo datos en JSON:
- Conectando con TRESTClient
- Obtener un resource TRESTRequest
- TRESTResponse: really, George?
- Si, también necesitas log en el cliente

De JSON a models

- ¿Por qué quieres hacer esto?
- JSONUtils, client side vs

Desarrollo multiplataforma

- Configuración del proyecto. Build groups
- Simuladores? Mejor Netflix :joy:
- Instalar PAServer
- Connection profile
- SDK Manager
- Cuidado con la memoria

Despliegue de la aplicación

- Provisioning
- Desarrollo vs vendor store
- Registro como desarrollador
- Obtención de certificados y otras torturas
- Desplegando ficheros de aplicacion
- Sandbox
- Controlar rutas en la aplicación

Notas sobre la arquitectura

- Mejor extender a MVC + Patrones (Repository, UseCase, UoW)
- Atención a la coordinación de hilos
- Aprovechar multitarea: `unit System.Threading;`
- Tener buen Log

Reglas y Procesos de Negocio.

- Necesitaremos buena arquitectura y buenas prácticas
- No solo hay métodos: aprovechemos las clases para distribuir la responsabilidad de cada pieza
- Desacoplemos con interfaces

Escalado

- Aprovechar el server-side
- Podemos desplegar multiples nodos servidor
- El cuello de botella sera la BD