

# Característica técnica del Winfra3

## Arquitectura

Winfra3 esta compuesto de tres capas:

1. Base de datos
2. Servidor de la aplicación
3. Aplicación del usuario (cliente) – cliente principal para Windows, cliente para Android

Servidor de aplicaciones implementa una serie de operaciones desde sencillas como “agregar nuevo cliente” hasta las mas larga y complicadas como “facturar el mes” o “cerrar el ejercicio contable”.

El cliente se comunica con el servidor y lanza operaciones – el cliente nunca accede directamente a la base de datos. El cliente no contiene ninguna funcionalidad que no sea introducir y presentar los datos.

## Comunicación cliente–servidor

Toda la comunicación esta encriptada utilizando SSL/TLS. La contraseñas no se almacenan ni se envían. Utilizamos funciones de hash para validar y almacenar las contraseñas.

Comprime la comunicación para mejorar el rendimiento del transporte. Esto es especialmente importante para dispositivos móviles.

La comunicación utiliza protocolo HTTP o HTTPS – funciona en cualquier tipo de configuración de red – local o internet. No requiere ningún tipo de acceso adicional como VPN o terminal server para conectar varias oficinas, puntos de ventas etc.

La comunicación esta diseñada para funcionar en las redes no fiables como wifi, 3g y permite que la red se desconecta y vuelve a conectar durante del uso de la aplicación. Permite reinicio de los servidores sin necesidad de salir de la aplicación o cerrar pantallas abiertas. La aplicación reconecta automaticamente sin intervención del usuario.

La interfaz del servidor esta implementada utilizando un estándar de comunicación común JSON (basado en ficheros de texto) para poder ejecutar las operaciones desde varios clientes programados en python, .NET, Java (Android) o JavaScript.

Gestión de errores esta implementada en el servidor, cliente controla la respuesta de servidor en cada llamada y presenta el error producido.

## Base de datos

Todas las operaciones de la base de datos se ejecutan como transacciones aisladas. Si la operación falla durante el procesamiento la base de datos esta devuelta automaticamente a su estado original.

Base de datos esta rigurosamente normalizada con control de integridad de los datos implementado automaticamente en todas las operaciones.

La base de datos esta diseñada para utilizar “identificadores universalmente únicos” UUID como identificadores de todos los registros de la base de datos. El diseño permite fusión automática de datos desde múltiples bases de datos como por ejemplo trabajar desde dispositivos desconectados como terminales de punto de venta o dispositivos móviles.

El diseño cuenta con la alta de múltiples empresas en la misma base de datos.

El diseño de la aplicación permite trabajar con varias bases de datos. Aunque el sistema esta optimizado para base de datos Postgres, el diseño permite cambio a otras base de datos.

## Servidor de aplicaciones

Multiplataforma – el servidor se puede ejecutar en los ordenador con sistema operativo Windows o Linux (Ubuntu, CentOS, RedHat). Funciona correctamente en los servidor virtualizados y en los contenedores.

La base de datos (Postgres), el servidor de aplicaciones (Apache) y el lenguaje de programación (Python) es de código abierto (open source) y se trata de software libre. Eso nos proporciona varias ventajas:

1. acceso a códigos fuente nos da poder analizar posibles problemas a nivel de la plataforma
2. no depender de un contrato comercial – no hay problema de final de vida del producto
3. no hay licencias para el software, no hay limite de numero de instalaciones, no hay versiones limitadas
4. el coste de obtención de software de la plataforma es 0€

El servidor permite ejecutar procesos de larga duración independiente del usuario. El usuario no tiene que esperar a que la operación finalice y puede apagar el programa/ordenador inmediatamente después de iniciar la operación.

El servidor mantiene el registro de todas las operaciones que se han ejecutado junto con datos de usuario que las ejecutó.

## Impresión e informes

Winfra implementa sus propias colas de impresión – cualquier dispositivo con winfra (ordenador windows, tablet android) puede imprimir en cualquier impresora desde cualquier despacho/oficina.

El diseño de los informes utiliza un sistema propio de Winfra que permite parametrizar los informes libremente y exportarlos en múltiples formatos – XPS, PDF o Excel.

## Seguridad

La seguridad de la plataforma se basa en minimizar la área del posible ataque. La típica instalación solo abre acceso al servidor de aplicación (ej. puerto 443). Ningún otro componente de la plataforma es accesible.

Todas las operaciones requieren que el usuario se autentique con la aplicación. Los credenciales aportados por el usuario se re-evalúan en cada operación.

El sistema de permisos se aplica a cada operación por separado. Es posible permitir o denegar la ejecución de cada botón o opción de menú por separado. Para evitar la complejidad de configuración el sistema ya viene pre-configurado con las opciones mas típicas.

El servidor controla la versión de los clientes en todo el momento y no permite la ejecución de las versiones antiguas. La medida asegura que todos usuarios siempre tienen instalada la versión correcta incluso si el usuario no cierra el programa.

## Clientes

El cliente para MS Windows es el cliente principal e implementa todas las opciones de Winfra.

Clientes para Android implementan varios aspectos de Winfra que requieren movilidad como preventa o reparto mecanizado.

La gestión de almacen esta implementada como cliente web adaptado para móviles/tablets.

## Instalaciones y actualizaciones

Una vez instalado el Winfra se actualiza con un único comando desde el servidor. El comando actualiza y reinicia los servicios. Además prepara el despliegue de la actualización de los clientes. Actualización de los clientes tiene varios niveles de automatización:

- automático – actualiza y reinicia el cliente sin atención del usuario. Este tipo de actualización utilizamos

para procesos automáticos como colas de impresión.

- automático con aviso – avisa al usuario que hay una nueva actualización y la instala automáticamente. Este es el proceso para los clientes de Windows.
- aviso – avisa al usuario pero le permite posponer la actualización. Esto es el proceso utilizado para dispositivos móviles para permitir al usuario trabajar con la versión anterior si no tiene conexión para descargar la actualización (ej. disponibilidad de wifi).

El proceso de actualización dura normalmente unos 2 minutos, incluyendo actualización del servidor y clientes. Todo el proceso de preparación de actualizaciones esta también automatizado. El equipo de Winfra puede prepara una nueva actualización desde una modificación del código en unos 15 minutos.

## Control de calidad

La aplicación ha sido diseñada desde principio para simplificar su continuo desarrollo y mantenimiento y para asegurar alto nivel de calidad.

De punto de vista de la arquitectura los principios mas destacados son:

- programación contra una fachada/interfaz – la fachada abstrae la funcionalidad interna del servidor y permite modificar la lógica u/e estructura de la aplicación sin tener que modificar los clientes
- diferenciamos los datos para almacenar y datos para enviar – para cada operación enviamos solo los datos necesarios en lugar de enviar el registro entero. Así aparte de mejoras importantes de rendimiento podemos modificar la estructura de los registros de base de datos sin tener que modificar los datos enviados
- abstracción de la base datos (ORM) – automatiza las consultas, mantiene solo un modelo de datos (no hay acceso a tablas y columnas) y nos permite utilizar diferentes bases de datos para pruebas y para producción.
- cada operación esta aislada y ejecutada independientemente de los de mas – implementamos cada operación como un nuevo fichero/objeto para minimizar impacto de los posibles errores a otras operaciones y poder seguir fácilmente los cambios
- gestión de errores automática durante de la ejecución asegura la consistencia de los datos en caso de errores imprevistos

De punto de vista de la metodología de desarrollo los principios son:

- control de modificaciones (versiones) de todos los ficheros incluidos en la aplicación, tanto servidor como cliente. Registramos autor, fecha y orden de todas las modificaciones, cada linea de código.
- pruebas de unidad/integración automatizadas – para cada operación existe correspondiente código de prueba que automáticamente ejecuta la operación y revisa sus resultados. Las pruebas ejecutamos después de cada modificación. En este momento (agosto 2019) existen 1.800 pruebas que cubren 90% del código (unas 250.000 lineas del servidor).
- actualizaciones automatizadas – proceso de preparar y actualizar nuevas versiones esta automatizado para evitar errores y nos permite instalar actualizaciones a diario
- control y seguimiento de las incidencias – reportadas por usuarios o detectadas internamente, las incidencias entran en la base de datos donde indicamos su estado y si han sido resueltas. Si la incidencia requiere cambio de código preparamos la correspondiente prueba para no volver a repetir el mismo error.
- herramientas para analizar el comportamiento de la aplicación en producción. Disponemos de análisis estadística del uso de aplicación para detectar cambios en rendimiento y respuesta. Optimizamos Winfra en base de números concretos, siempre sepamos la mejora que estamos consiguiendo.
- winfra3 dispone del servicio de envío y análisis de los informes de errores. El servicio permite a nuestro equipo de atención al cliente revisar detallados informes de errores en un único sitio unificando todas las plataformas: servidor, cliente windows y cliente móvil.

## Implementación de los servicios

Servicios que conforman la instalación completa de winfra. Incluyo servicios programados por winfra y servicios de terceros.

### Cliente

Existen varios clientes:

- Principal cliente de winfra - MS Windows .NET 4.0., interfaz de usuario WPF. Instalación XCOPY.  
Hay que instalar runtime de .NET 4.0 en algunas versiones de XP, incluido en nuevos windows. Es compatible con Windows 2003, XP, 7, Vista, 8, 10.  
Existe versión de cliente para TPV y para servidor de impresión con los mismo requerimientos.
- Cliente para preventa y reparto mecanizado – aplicación para Android 6, compatible desde 4.1
- Mecanización de almacén – aplicación web compatible con todos navegadores modernos de móviles, tablets o PC.

Todos los clientes requieren conexión por HTTP/HTTPS al servidor. El cliente para preventa y reparto mecanizado necesita la conexión para cargar y descargar de los datos. Toda la impresión de ls documentos en el reparto mecanizado se ejecuta en el dispositivo sin necesidad de servidor.

### Servidor

La instalación recomendada es en el servidor linux Ubuntu 18.04. Esta probado con Ubuntu 12.04, 14.04, 16.04, CentOS 7, RedHat 7.6 y en Windows Server 2003, 2012.

Winfra funciona con instalación en maquina virtual (utilizamos XENServer) y con contenedores alojados en centros de datos (probado con 1&1 y Clouding.io).

Los servicios instalados son:

- Apache 2.4 – servidor de la aplicación con el modulo ModWSGI para ejecución de python
- Redis – servicio de colas de trabajos de larga duración.
- Memcached – servicio de cache en memoria para cachear resultados de tareas, informes y para optimizar el rendimiento.
- Celeryd – servicio de ejecución de las tareas de larga duración
- Postgresql – servicio de base de datos

Todos los servicios están normalmente instalados en la misma maquina, pero por razones de rendimiento y/o protección contra fallos es posible separarlos en varias maquinas. Por ejemplo dedicar una maquina solo a la base de datos. Todos los servicios comunican por TCP/IP.

### Rendimiento

La aplicación permite balanceo de carga, incluido el mas sencillo round-robin sin afinidad de servidor. (Winfra no mantiene sesiones.) Aunque según nuestra experiencia el cuello de botella suele ser en la base de datos.

Disponemos de heramientas para analizar el rendimiento del servidor y para analizar las conexiones de red para poder detectaar problemas con redes wifi etc.