

Whitepaper:



DESPLIEGUE APLICACIONES DE POWERBUILDER EN LA NUBE

Última actualización: 21 de julio de 2020

APLICACIONES DE ESCRITORIO EN LA NUBE

La nueva generación de PowerServer (PowerServer 2021) le permite implementar automáticamente proyectos de PowerBuilder como "aplicaciones de escritorio en la nube". Como lo indica el nombre, tales aplicaciones cuentan con atributos de "escritorio" y "nube". **Escritorio** lo que significa que la aplicación seguirá ejecutándose de forma nativa en el escritorio de Windows con la potente funcionalidad y la rica interfaz de usuario de una aplicación de escritorio. **Nube** es decir, la base de datos y las operaciones relacionadas con la base de datos que residen en la nube (en lugar de en local) con despliegue automático del cliente a los usuarios a través de Internet. En otras palabras, puede pensar en las "aplicaciones de escritorio en la nube" como la aplicación de escritorio moderna que requiere conexión a Internet para instalar y ejecutar.

POWERSERVER 2021 – CAMBIOS

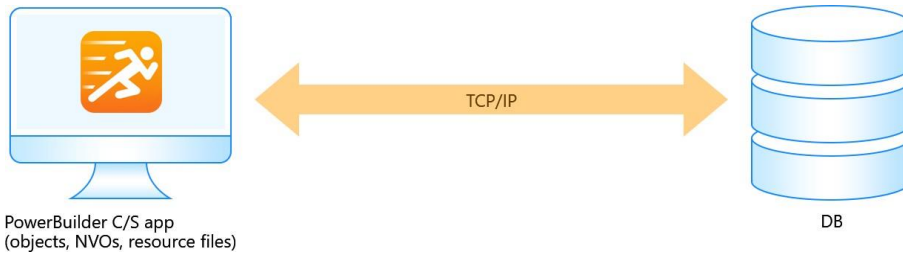
PowerServer 2021 reemplaza la implementación de Web browser de la generación anterior de PowerServer con una implementación de aplicación de escritorio en la nube. Este reemplazo de la implementación anterior es mucho más poderoso y ofrece las siguientes ventajas clave.

- Esencialmente, soporta el 100% de las funciones de PowerBuilder. Esto ahora es posible porque PowerServer 2021 ya no se traduce a JavaScript (por ejemplo, la aplicación web) y en su lugar usa PowerScript de forma nativa (por ejemplo, la aplicación de escritorio).
- Las aplicaciones de escritorio en la nube son aplicaciones que se pueden instalar en Internet. Inicialmente, se instalará una aplicación de escritorio en la nube accediendo a la URL de la aplicación en un navegador web. Después de eso, la aplicación no depende de un navegador web (tipo, versión o configuración), sino que se ejecutará y actualizará según sea necesario a través de Internet.
- La comunicación entre el escritorio y el servidor se realiza a través de la interfaz REST / JSON estándar (en lugar del formato binario propietario). Como se trata de REST / JSON estándar, se puede proteger utilizando cualquier enfoque que desee, por ejemplo, OAuth 2.0 para autenticación de usuario y cifrado AES para proteger los datos JSON.
- .NET Core ahora es compatible. Esto cambia el panorama para .NET de 3 formas principales: 1) admite cross-OS (por ejemplo, servidor Linux), 2) el rendimiento del lado del servidor es más rápido y 3) Microsoft rediseñado desde cero para ser implementado en la nube (incluyendo compatibilidad con Docker y versiones side-by-side).
- El mecanismo de activación de la licencia del producto ha sido revisado para compatibilidad con la nube.

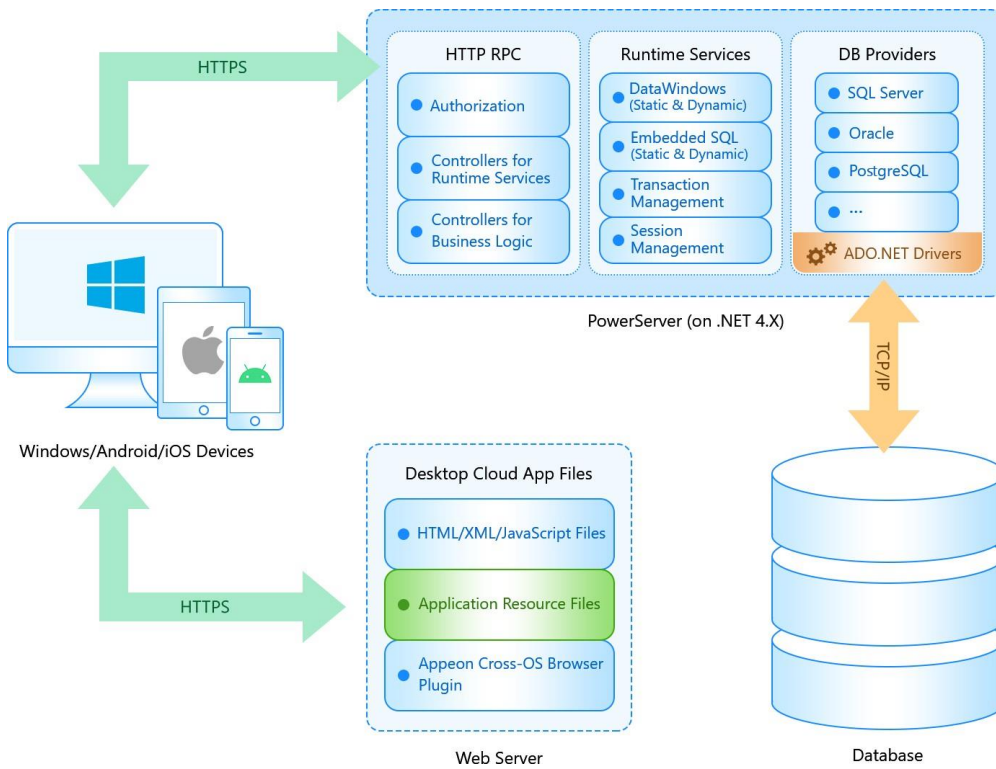
La desventaja clave de este cambio en la implementación de la aplicación de escritorio en la nube es que PowerServer 2021 ya no puede admitir dispositivos iOS y Android. Esto se debe a que PowerScript solo es compatible con el sistema operativo Windows.

COMPARACIONES DE ARQUITECTURA DE POWERSERVER

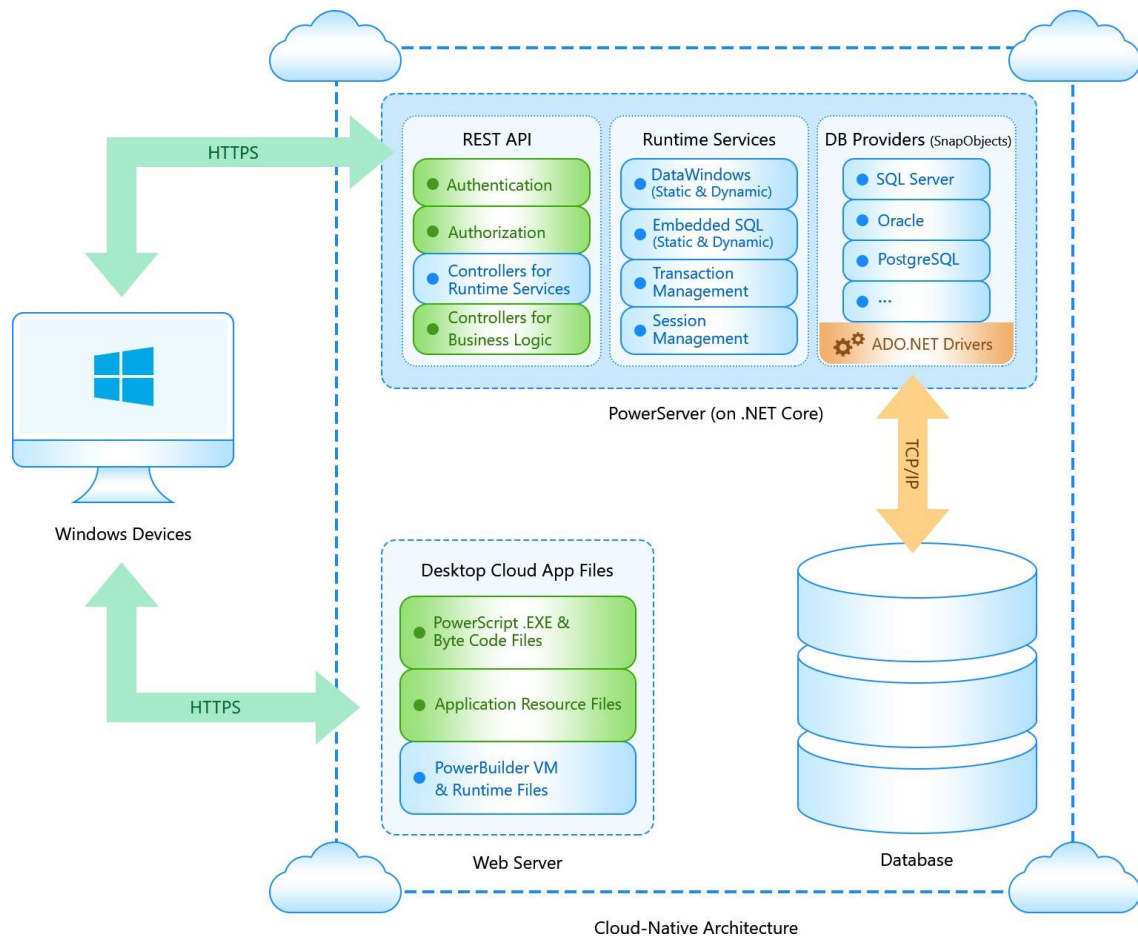
Arquitectura de cliente / servidor PowerBuilder:



Arquitectura anterior de PowerServer:



Arquitectura PowerServer 2021:



¿Cómo funciona PowerServer 2021?

- PowerServer 2021 adopta el estándar abierto REST API, reemplazando el RPC HTTP propietario anterior, para la comunicación entre el cliente y los servicios de runtime de PowerServer (por ejemplo, ejecución de DataWindow, ejecución de SQL incorporado, etc.).
- El lado del servidor ahora es mucho menos un black box. Específicamente, los desarrolladores pueden ver y modificar el código fuente de prácticamente toda la capa REST API (resaltada en color verde), como implementar su propia autenticación y autorización.
- Implementa una arquitectura nativa de la nube para que pueda ir más allá del alojamiento básico y aprovechar las tecnologías de la nube, como Docker / Kubernetes.
- Adopta el marco de .NET Core de código abierto (en lugar del marco de .NET antiguo) y el marco de ORM de SnapObjects, que mejora el rendimiento del runtime y admite el sistema operativo cruzado (por ejemplo, Windows, Linux, etc.).
- Es posible configurar el equilibrio de carga entre varias instancias de PowerServer. Bajo dicha configuración, se requieren sesiones fijas para preservar la integridad de la transacción. Específicamente, todas las solicitudes en una transacción serán manejadas por la misma instancia del servidor.

DESPLIEGUE DE LA APLICACIÓN DE ESCRITORIO EN LA NUBE

Compilación de aplicaciones

Tradicionalmente, cuando compila una aplicación PowerBuilder utilizando el p-code, PowerBuilder crea bibliotecas dinámicas (archivos PBD). Los archivos PBD contienen todo, como objetos visuales, DataWindows / DataStores, código PowerScript y SQL incorporado. Todos los archivos PBD deben instalarse en el escritorio y trabajar juntos para que la aplicación se ejecute.

La compilación de aplicaciones de escritorio en la nube es diferente del enfoque tradicional de p-code:

- El acceso a los datos ahora se ejecuta en el servidor en lugar del escritorio:
 - DataWindows / DataStores estáticos se convierten automáticamente a modelos .NET y son desplegados en el servidor durante el tiempo de diseño.
 - Dynamic DataWindows / DataStores se crearán en el escritorio y se enviará su SQL como cadenas al servidor durante el tiempo de ejecución.
 - Las operaciones de datos de todos los DataWindows / DataStore (sin importar estáticas o dinámicas) son realizado en PowerServer mediante la llamada a las API REST pertinentes.
 - Los SQL integrados se implementarán como cadenas en el servidor durante el tiempo de diseño.
 - Los SQL dinámicos se crearán en el escritorio y se enviarán como cadenas al servidor durante el tiempo de ejecución
 - La ejecución de todos los SQL (sin importar incrustados o dinámicos) se realiza en el PowerServer mediante la llamada a las API REST pertinentes.
- Todos los PowerScript y todos los demás objetos (además de DataWindows / DataStores) continúan ejecutándose en el escritorio. En otras palabras, PowerServer no divide automáticamente la lógica empresarial de la aplicación escrita en PowerScript.
- Aunque no es necesario dividir la lógica empresarial para las aplicaciones de escritorio en la nube, si desea reutilizar la lógica empresarial de la aplicación en otro lugar u optimizar el rendimiento general de la aplicación, [PowerScript Migrator](#) El producto puede automatizar en gran medida la conversión de dicha lógica empresarial a las API REST / JSON.
- Todos los archivos PBD se desglosan muy granularmente en cada objeto individual / archivo de definición. Por ejemplo, cada archivo SRW, SRD, SRU, etc. tendría su archivo de p-code correspondiente individual (que tiene nuevas extensiones de archivo, como .dwo, .apl, .fun, .win, .udo) en lugar de un monolítico Archivos PBD. Luego, cada versión de cada archivo de p-code tiene un identificador único, de modo que cuando un usuario ejecuta la aplicación en la nube de escritorio, los archivos de p-code se pueden descargar y actualizar gradualmente, dependiendo de la ventana y la funcionalidad a la que esté accediendo.

Por lo tanto, verá que, con el nuevo enfoque de compilación, está mejorando la compilación de PowerBuilder no solo para admitir un modelo de cliente / servidor, sino también para admitir un modelo en la nube. En pocas palabras, el acceso a los datos se mueve automáticamente desde el escritorio a la nube, y la instalación de la aplicación se rediseña para funcionar de manera eficiente a través de Internet, proporcionando los cambios clave necesarios para implementar proyectos existentes de cliente / servidor en la nube.

Despliegue de aplicaciones

Implementación en el servidor web

Después de la compilación de la aplicación, los siguientes dos tipos de archivos se implementarán en el servidor web:

- Archivos de p-code que son de tamaño granular (no el formato PBD monolítico).
- El archivo .exe de la aplicación y los archivos de soporte como archivos de recursos, archivos OCX y archivos de runtime de escritorio (por ejemplo, PBVM). Dichos archivos son los mismos que en las aplicaciones PowerBuilder de cliente / servidor. Habrá ajustes de configuración dedicados para que usted especifique estos archivos de soporte que se empaquetarán e implementarán con la aplicación en el escritorio.

Aunque se utiliza un servidor web, técnicamente está actuando como un servidor de archivos (es decir, no se ejecuta ningún código en el servidor web).

Implementación en PowerServer

Los siguientes dos tipos de archivos se convierten desde DataWindows / DataStores / SQL incorporado, y se implementan en PowerServer, se compilan junto con las DLL de servidor necesarias y se publican como aplicaciones .NET Core (API REST).

- Modelos .NET;
- SQLs.

Despliegue del cliente

Las aplicaciones de escritorio en la nube son aplicaciones que se pueden instalar en Internet. Inicialmente, se instalará una aplicación en la nube de escritorio accediendo a la URL de la aplicación en un navegador web. Después de eso, la aplicación no depende de un navegador web (tipo, versión o configuración), sino que se ejecutará y actualizará según sea necesario a través de Internet.

TAREAS MANUALES CLAVE

PowerServer es una solución de implementación en la nube altamente automatizada que requiere muy pocos cambios de código. Sin embargo, hay algunas tareas clave que no se manejan automáticamente y que requieren un esfuerzo de programación manual:

- Autenticación de usuario. Es posible que las aplicaciones de escritorio en la nube apliquen la autenticación OAuth2.0. Se proporcionarán algunas configuraciones después de la implementación de la aplicación para configurar la estrategia de autenticación de usuario para la aplicación.
- Conexión a la base de datos. La configuración de SQLCA se configurará para PowerServer como configuración de fuente de datos y configuración de conexión de base de datos dinámica, que establece la conexión entre PowerServer y la base de datos.