Cómo Realizar el Upgrade de 6.X a Liferay Digital Experience Platform

Liferay

Índice

Introducción 1
Cronograma General del Upgrade 1
Consideraciones importantes: 1
Cronogramas Generales 3
Revisión de Arquitectura: 1-2 semanas. 3
Revisión de escalabilidad: 1-2 semanas 3
Upgrade de Liferay: 1-4 meses 3
Pruebas: 1-2 semanas 3
Ajuste de Desempeño: 1-2 semanas 3
Cambios en la Infraestructura 4
Matriz de Compatibilidad 4
Búsqueda 4
JDK 5
Plan de Implementación 5
Upgrade del Bando de Datos 6
Antes de Comenzar 6
Herramienta de Actualización del Banco de Datos . 6
Resolución de Problemas 7
Actualización del Núcleo Isolado 8
Actualizando los Módulos 8
Actualizando su Código 10
Liferay Project SDK 10
Liferay Workspace 11
Blade CLI 11
Principales Cambios 12
Liferay Developer Studio 13
Code Upgrade Tool 13
Actualizando los Plugins SDK para Liferay Workspace 13
Ubicando los Cambios 14
Actualizando Customizaciones en la Nueva Estructura de Módulos 15
Escenarios de Actualización de Código 15
Migrando un 6.2 WAR para un WAR soportado por Liferay DXP 15
Convirtiendo un portlet en un módulo OSGI . . 16
Actualizando Temas 18
Recursos Adicionales 19
Resumen 20
Siguiendo Adelante 20

Introducción

Como la demanda de experiencias digitales en todo momento y en todas partes ha aumentado, la necesidad de contar con la tecnología adecuada para ofrecer estas interacciones y permanecer ágil también aumentó. Liferay Digital Experience Platform permite a las empresas digitales administrar y proporcionar experiencias de cliente consistentes y conectadas a través de puntos de contacto digitales, incluyendo dispositivos móviles, desktop, kioskos, dispositivos inteligentes y mucho más.

El upgrade a Liferay DXP es una inversión para satisfacer la creciente necesidad de la experiencia digital, al mismo tiempo que establece la base para atender al público digital cada vez más conectado. La actualización coloca a su empresa en la mejor posición para aprovechar los últimos avances de Liferay para negocios digitales, incluyendo headless APIs, auto-tagging de documentos, recomendaciones de contenido, segmentación avanzada, personalización y mucho más.

Este documento tiene como objetivo definir el camino recomendado por Liferay para realizar el upgrade de la plataforma en su organización. Una gran actualización de tecnología requiere un análisis profundo de los requisitos de su negocio, planificación cuidadosa, pruebas y ejecución para obtener éxito. Antes de iniciar cualquier etapa, el equipo de Global Services Liferay, nuestro equipo de consultores profesionales con amplia experiencia en actualizaciones de la plataforma Liferay en diversos escenarios, puede ayudarle en un análisis crítico de sus necesidades a través del Liferay Upgrade Analysis Program. Combinando este análisis completo con las principales consideraciones necesarias en una actualización para el Liferay DXP, usted puede tomar decisiones más asertivas para que su empresa aproveche al máximo todos los recursos de la plataforma Liferay.

Cronograma General del Upgrade

Consideraciones importantes:

Upgrades requieren modificaciones en datos, códigos e infraestructura. El proceso es naturalmente complejo. Algo que debe tenerse en cuenta es ***lo que se desea versus lo que es realmente necesario.***

Se debe dar tiempo al usuario para realizar pruebas de desempeño para detectar cualquier conflicto o defectos relacionados con la codificación (esto incluye posibles defectos o errores en Liferay DXP). Esto debe ocurrir antes de que cualquier ajuste de ambiente sea realizado.

Se debe tener en consideración el tiempo extra de revisar toda la personalización, hooks, temas, tablas, plugins, etc. Cuanto más compleja sea la configuración, más tiempo será necesario para planear posibles problemas. En Liferay DXP, todas las propiedades del portal deben ser revisadas para determinar la necesidad real de la funcionalidad. Los clientes deben hablar con el equipo de Liferay Support para aclarar cualquier incertidumbre sobre lo que una propiedad de portal hace.

Responder estas preguntas a continuación pueden ayudar a definir un calendario y las expectativas para administrar mejor los riesgos.

1. **¿Cuál es la versión que utilizo?** Si utiliza algunas versiones más antiguas que la versión actual de Liferay, recuerde que sus datos necesitarán pasar por el camino completo de actualización de versiones anteriores hasta que se actualice a Liferay DXP. Esto significa que si está en Liferay Portal 6.1, sus datos serán actualizados primero a 6.2 antes de ser actualizados a Liferay DXP. Recomendamos usar la herramienta de Upgrade para hacerlo, ya que puede manejar las complejidades involucradas. El único caso de uso que debe actualizar manualmente es cuando la actualización del Portal es de una versión anterior a la 6.1.
2. **¿Cuál es el tamaño de mi base de datos?** Si su proyecto tiene muchos datos es esencial tener una base de datos indexada correctamente. Además, reserve más tiempo si usted tiene una base de datos mayor.
3. **¿Necesitaré hacer el upgrade para los contenido web, plantillas y estructuras?** El contenido web, las plantillas y las estructuras se pueden actualizar a 7.x. Pero si tiene la versión 6.1, puede tener algunos problemas con el requisito de tener elementos únicos en una estructura. Más información se puede encontrar aquí.
4. **¿Cuántos portlets son necesarios realizar el upgrade?** No todos los portlets necesitarán ser actualizados a Liferay DXP. Un análisis detallado puede traer una estimación de cuánto tiempo será necesario y cuántos desarrolladores deben dedicarse a este proceso.
5. **¿Tendré que sustituir muchos JSP?** JSP han cambiado mucho desde las versiones 6.1 y 6.2. Le recomendamos que no sustituya los JSP completamente, de ser posible. Algunos JSP tienen extensiones que usted puede conectar sin recurrir a eso.
6. **¿Tengo un EXT para el upgrade?** La buena noticia es que con Liferay DXP creamos mucho más puntos de extensión, entonces el EXT finalmente puede ser removido.
7. **¿Necesito convertir para bundles OSGi?** Esto llevará más tiempo, pero la inversión vale la pena.

Cronogramas Generales

Revisión de Arquitectura: 1-2 semanas

La revisión de arquitectura interna de un proyecto generalmente tarda 1-2 semanas. Los desarrolladores deben ser conscientes de la nueva arquitectura modular y lo que esto significa, ya que a veces no ven la implicación de ello en el momento de la configuración de sus dependencias.

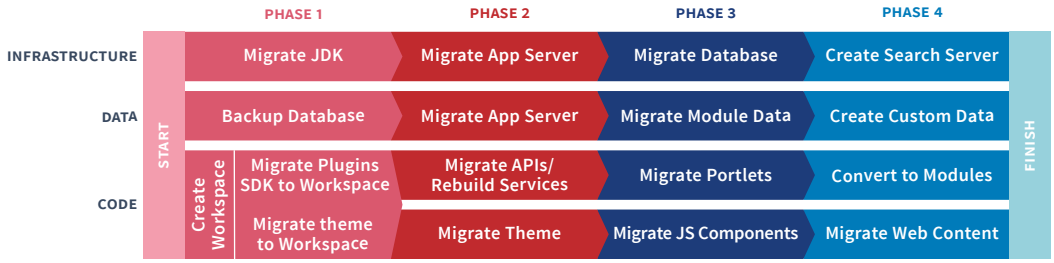
Revisión de escalabilidad: 1-2 semanas

Después de la finalización de la Revisión de Arquitectura, 1-2 semanas serán necesarias para determinar si el diseño del código actual considera la escalabilidad necesaria para su caso de uso. Esto se debe hacer con su revisión de código.

Upgrade de Liferay: 1-4 meses

Este documento le guiará para el Upgrade de Liferay. En esta fase usted realizará los cambios necesarios para actualizar sus datos, personalizaciones y ambientes para Liferay DXP 7.1. La revisión de escalabilidad y arquitectura debe asegurar que usted utilice de la mejor forma el calendario de actualización.

Ejemplo de Etapas para el Upgrade a Liferay DXP



Pruebas: 1-2 semanas

Este es el período en que el código se debe corregir para llegar al paso de ajuste de desempeño y la prueba de carga. Puede comprobar la arquitectura mínima, además de las plataformas/requisitos de hardware en nuestro documento Performance de Liferay DXP.

Ajuste de Desempeño: 1-2 semanas

Esta fase se inicia cuando se resuelven los cuellos de botella relacionados con el código y se deben cambiar los archivos de configuración para descubrir los mejores valores de desempeño.

El intento de ajustar el desempeño antes de la revisión de código puede llevar a los cuellos de botella en la actualización.

Durante esse período, recomendamos que lea Deployment Checklist do Liferay DXP.

Además, todos los otros sistemas (base de datos, Apache HTTPd, servidor de aplicaciones, Liferay DXP en sí) necesitarán pasar por una revisión completa con la ayuda de los expertos de su equipo. El ajuste de desempeño tarda de 2 a 4 semanas en promedio con un equipo experimentado de QA o DBA. El tiempo puede variar dependiendo del conocimiento de su equipo.

Cambios en la Infraestructura

Matriz de Compatibilidad

Ahora que está listo para comenzar el upgrade, empiece a verificar si su ambiente está actualizado con la Matriz de Compatibilidade de Liferay DXP. Muchos ambientes ya no se soportan desde que se lanzó el 6.2. Es esencial que su sistema se ejecute en un ambiente compatible para que pueda recibir el soporte adecuado.

Búsqueda

El mayor cambio que usted notará es que la plataforma Liferay ahora requiere que el motor de búsqueda se ejecute como una JVM independiente, ya sea en el mismo servidor o en un servidor físico / virtual dedicado.

Como la búsqueda es una parte vital de cualquier implementación exitosa, mejoramos considerablemente la funcionalidad de búsqueda en el Liferay DXP. El perfil de desempeño y escalabilidad de un motor de búsqueda es drásticamente diferente si se compara al de un portal que transmite activamente impresiones de la web. En el pasado, Liferay Portal soportaba un mecanismo de investigación de Lucene o un motor de búsqueda de Solr implementado remotamente. Con Liferay DXP, continuaremos ofreciendo el Solr como un motor de búsqueda remoto compatible. Sin embargo, la preferencia es utilizar el Elasticsearch. Esto permite que el mismo mecanismo se utilice en el modo incorporado durante el desarrollo y en un modo independiente para la producción.

Si usted estaba usando anteriormente Solr, sólo tendrá que asegurarse de que está utilizando el módulo más reciente de Solr y actualizar a 7.x. Por favor, consulte nuestra documentación de búsqueda para configurar correctamente su servidor de búsqueda. Observe también que puede ejecutar el servidor de búsqueda en el mismo servidor, pero separar la JVM, en el caso que tenga recursos limitados, aunque no es lo recomendado.

JDK

Otro cambio en la matriz de compatibilidad es que utilizamos ahora el JDK11. Todos los servidores de aplicaciones ya son compatibles con JDK8 en la matriz de compatibilidad. Asegúrese de que su servidor de aplicaciones esté correctamente instalado.

Plan de Implementación

Con la introducción del contenedor OSGi en Liferay DXP, su plan de implementación necesita ser cambiado, ya que ahora hay algunas maneras diferentes de implementar sus plugins.

- **Bundles/módulos OSGi** son un nuevo tipo de plugins en Liferay DXP. Sólo son Java JARs simples con metadatos OSGi. Los paquetes OSGi sólo se pueden implementar en el contenedor OSGi y no pueden acceder a los servicios implementados como WAR, además de los servicios principales de Liferay. Este es el abordaje recomendado para la mayoría de los nuevos desarrollos y será el abordaje que Liferay considera para todos los nuevos desarrollos.
- **WARs** son aplicaciones web tradicionales de plugins Liferay (por ejemplo, *-theme.war, * -portlet.war, * -web.war). Los WAR no se benefician de los modelos explícitos de dependencia y gestión de ciclo de vida proporcionados por OSGi, pero afortunadamente en Liferay DXP, todos los WAR (excepto un EXT) se convierten en un WAB. Cuando implementa un archivo WAR en Liferay, se convierte automáticamente en un WAB.
- **WABs** son los bundles de archivos de la web. Esto le dará todos los beneficios de un bundle sin realizar la conversión. Este es el abordaje recomendado para implementar aplicaciones JavaEE (por ejemplo, SpringMVC o JSF), aplicaciones heredadas creadas para versiones anteriores de Liferay y temas.

Todos los plugins Liferay (excepto el EXT) ahora son implementados en el container OSGi. Este será o un bundle OSGi o uno WAB.

CUIDADO: Usted nunca debe implementar WAR, módulos, WAB de Liferay directamente usando las herramientas de implementación de su servidor de aplicaciones.

Al implementar bundles OSGi en el clúster administrado de un servidor de aplicaciones (por ejemplo, el modo de dominio de JBoss, WebLogic con el gestor de nodos, WebSphere con Deployment Manager), tendrá que contar con el Cluster Deployment Helper de Liferay. Esta herramienta utilizará los contenidos y códigos implementados y los reunirá en un archivo WAR. Cuando el servidor de aplicaciones implementa e inicia el WAR, los mecanismos de arranque colocarán los módulos OSGi en la carpeta de implementación OSGi de Liferay Digital Enterprise en el servidor de aplicaciones.

Upgrade del Bando de Datos

Antes de Comenzar

Ahora que su infraestructura ha sido actualizada para versiones compatibles, podemos concentrar nuestra atención en los datos. La primera cosa que debemos hacer es garantizar una copia de seguridad adecuada en caso de fallas.

Después, asegúrese de que está utilizando el algoritmo de permiso 6 si utilizas la versión 6.1. Si no está en el algoritmo de permiso 6, debe migrar a ese algoritmo de permiso. Puede encontrar información sobre cómo migrar algoritmos de permisos aquí.

Además, recuerde que todos los índices de la base de datos se han aplicado correctamente. Un índice ausente puede causar un retraso en el proceso de actualización. Si está ejecutando una actualización lenta, posteriormente puede que desee volver y agregar índices temporales adicionales para ayudar a acelerar el proceso.

Si usted fuera a staging, recomendamos que publique todo el contenido antes de ejecutar la actualización.

También recomendamos que usted deshabilite la indexación en la búsqueda durante el proceso. Para eso, usted debe agregar un archivo llamado `com.liferay.portal.search.configuration.IndexStatusManagerConfiguration.cfg` en tu `osgi/configs/folder` con el siguiente contenido:

```
indexReadOnly=true
```

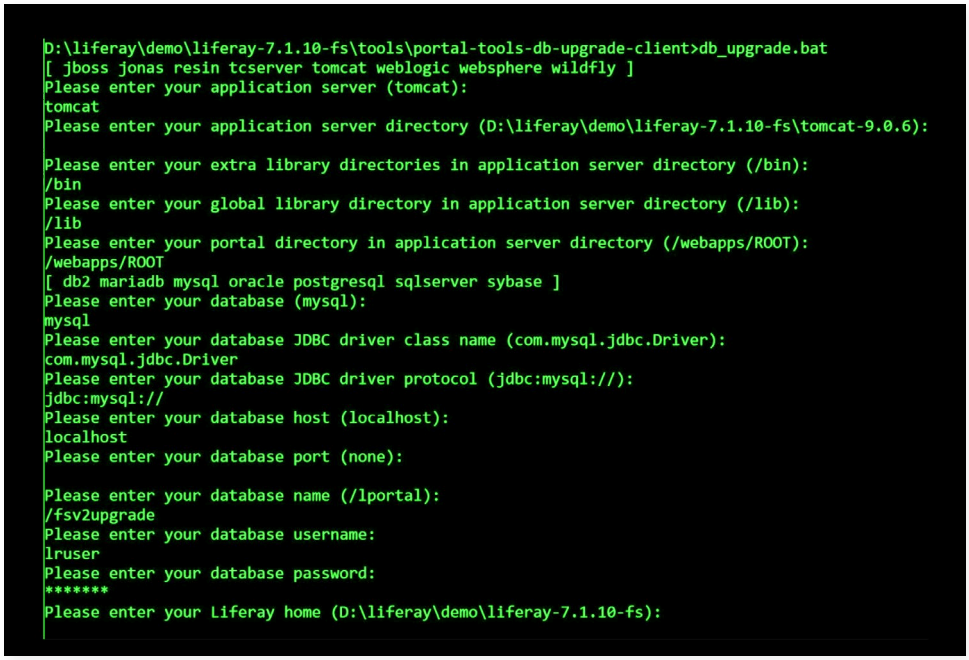
Al hacer esto, evitará la indexación y reducirá el tiempo de actualización. Después de actualizar su portal, defina la propiedad como falsa para poder indexar todos los objetos del panel de control.

Herramienta de Actualización del Banco de Datos

En Liferay DXP, las actualizaciones del banco de datos fueron transferidas para una herramienta independiente. Para ejecutar la herramienta:

```
db_upgrade.sh or db_upgrade.sh
```

La actualización requiere que tres archivos sean configurados antes de ejecutar: `app-server.properties`, `portal-upgrade-database.properties`, `portal-upgradeext.properties`. Usted puede hacer eso manualmente o la herramienta lo podrá ayudar.



La actualización de datos ahora está dividida en dos partes. La actualización básica es similar a lo que usted vio en el pasado. La siguiente parte actualizará los módulos OSGi. De forma predeterminada, la herramienta de actualización de base de datos se configura para actualizar ambos automáticamente.

Resolución de Problemas

Si su actualización se ha realizado correctamente, puede omitir esta sección. Si tiene problemas, revise algunos consejos para ayudarle.

- Si usted está tratando de realizar la actualización y ha encontrado algunos problemas, recuerde los consejos que presentamos antes de empezar. **La mayoría de los problemas en una actualización se producen debido a los datos dañados.** La única manera de resolver problemas de este tipo es corregir y eliminar los datos dañados. Afortunadamente, en Liferay DXP 7.1 y posteriores, puede reanudar la actualización desde donde se detuvo.
- Un problema común, pero fácil de resolver, con la actualización es la **configuración de propiedades que tienen errores tipográficos**. Si la actualización no se conecta a la base de datos o no encuentra los archivos correctos en la instalación de Liferay, verifique los archivos de propiedades.

Uno de los avances para Liferay DXP es la separación entre un “núcleo” lógico y una serie de “módulos”. Esto nos permite hacer pequeñas actualizaciones incrementales para ayudar en la selección y resolver problemas relacionados con los datos.

Actualización del Núcleo Isolado

NOTA: Aislar y actualizar el núcleo del portal y los módulos se deben hacer sólo como parte de depuración, creación de scripts de limpieza de datos y pruebas en un ambiente de no producción. En general, se recomienda probar la actualización con respecto a una copia de su base de datos de producción y resolver todos los problemas antes de intentar realizar una actualización completa en la producción.

Usted puede configurar el portal para apenas actualizar el núcleo, y no los módulos, al agregar un campo llamado `com.liferay.portal.upgrade.internal.configuration.ReleaseManagerConfiguration.cfg` en el `osgi/configs/folder` con el siguiente contenido:

```
autoUpgrade=false
```

Además, usted debe verificar el proceso de ejecución a través de `portal.properties`:

```
verify.frequency=0
```

Se puede reactivar el proceso de verificación después que todas las actualizaciones hayan sido realizadas con éxito.

Si los problemas se producen durante la actualización de los módulos, debe configurar la herramienta de actualización para ejecutar manualmente la actualización de los módulos. Complete el upgrade del núcleo y haga una copia. Si tiene problemas, puede reiniciar desde allí. Los módulos OSGi se pueden actualizar individualmente. Ofrecen copias seguras si se ejecutan correctamente.

Para las técnicas más avanzadas, puede intentar utilizar el depurador. La ventaja de esto es que, en algunos casos, puede corregir los datos dañados en la memoria y se guardarán en la base de datos, corrigiendo sus datos sin reiniciar. Esto requiere un alto nivel de conocimiento sobre las tablas y sólo se recomienda para los desarrolladores más experimentados en Liferay. Esta técnica también abre una nueva opción durante las actualizaciones de módulo, ya que los paquetes configurables de OSGi se actualizan en etapas. Puede utilizar un depurador para interrumpir en cualquier paso y tomar una copia de su base de datos allí.

Actualizando los Módulos

Para realizar la actualización de los módulos, puede utilizar el Gogo shell. Conéctese al shell al ejecutar el `telnet localhost 11311`. Utilice los comandos disponibles en el espacio de nombres actualizado. Por ejemplo:

```
upgrade:list
upgrade:execute
upgrade:check
verify:list
verify:execute
```

Al introducir `upgrade:list`, la consola mostrará los módulos disponibles para la actualización siempre que todas las dependencias de actualización estén contempladas.

Si ningún módulo aparece, será necesario actualizar primero sus dependencias. Usted puede realizar el comando `scr:info {upgrade_qualified_class_name}` para verificar cuáles dependencias no están contempladas. Por ejemplo: `scr:info com.liferay.journal.upgrade.JournalServiceUpgrade`

Al teclear: `list {module_name}`, la consola mostrará el paso a paso que necesita para actualizar los módulos. Para entender cómo funciona esto, se recomienda ver un ejemplo; Si ejecuta aquel comando para módulos de servicio favorables, verá esto:

```
Registered upgrade processes for com.liferay.bookmarks.service 1.0.0
  {fromSchemaVersionString=0.0.1, toSchemaVersionString=1.0.0-step-3,
upgradeStep=com.liferay.bookmarks.upgrade.v1_0_0.UpgradePortletId@497d1106}
  {fromSchemaVersionString=1.0.0-step-1, toSchemaVersionString=1.0.0,
upgradeStep=com.liferay.bookmarks.upgrade.v1_0_0.UpgradePortletSettings@31e8c69b}
  {fromSchemaVersionString=1.0.0-step-2, toSchemaVersionString=1.0.0-step-1,
upgradeStep=com.liferay.bookmarks.upgrade.v1_0_0.UpgradeLastPublishDate@294703b6}
  {fromSchemaVersionString=1.0.0-step-3, toSchemaVersionString=1.0.0-step-2,
upgradeStep=com.liferay.bookmarks.upgrade.v1_0_0.UpgradeClassNames@7544b6e5}
```

Esto significa que hay un proceso disponible para actualizar los favoritos de la versión 0.0.1 a 1.0.0. Para completarlo, usted necesitaría ejecutar cuatro pasos y el primero es aquel que se inicia en la versión inicial y termina en el primer paso de la versión de destino (el mayor número de etapa, paso 3 para este ejemplo), UpgradePortletId en este caso. La última etapa es aquella que se inicia en el último paso de la versión de destino (el número más pequeño del paso, el paso 1) y termina en la versión de destino (1 0 0), UpgradePortletSettings, en este caso.

Al introducir upgrade: `execute {module_name}`, usted actualizará un módulo. Es importante tener en cuenta que si hay un error durante el proceso, puede reiniciar el proceso desde el último paso ejecutado exitosamente, en vez de ejecutar todo el proceso de nuevo. Usted puede ver el estado de su actualización ejecutando `upgrade: list {module_name}`.

Al introducir `` `upgrade : check `` en Gogo shell, se mostrará los módulos que no llegaron a la versión final. Así, usted tendrá una manera de identificar las fallas en el upgrade al final del proceso.

Para entender cómo funciona este comando, por favor considere este ejemplo: imagine que la actualización del módulo `com.liferay.dynamic.data.mapping.service` falló en el paso 1.0.0-step-2. Si ejecuta el comando `upgrade:check` usted tendrá:

```
Would upgrade com.liferay.dynamic.data.mapping.service from
1.0.0-step-2 to 1.0.0 and its dependent modules
```

Esto significa que tendrá que corregir el error y ejecutar la actualización a este módulo una vez más. Tenga en cuenta que los módulos dependientes de `com.liferay.dynamic.data.mapping.service` deben actualizarse una vez que el primero sea actualizado correctamente.

Además, puede ejecutar el proceso de comprobación desde la línea de comandos utilizando `verify:list`. Esto muestra todos los procesos de verificación disponibles. Añada `verify:execute {verify_qualified_name}` para ejecutarlo.

Actualizando su Código

Finalmente, vamos a abordar cómo actualizar su código para Liferay DXP. Si tiene varios miembros en su equipo, esta parte se puede iniciar junto con la actualización de datos. Para actualizar su código, primero deben familiarizarse con el Liferay Project SDK y con las herramientas contenidas en él.

Liferay Project SDK

Como parte del DXP, Liferay ha invertido fuertemente en la mejora y modernización de sus herramientas para desarrolladores. El Liferay Plugin SDK *ant-based* se ha sustituido en el nuevo Liferay proyecto SDK, un enfoque moderno que incluye Gradle o Maven que proporcionan una experiencia más completa, de extremo a extremo.

El Liferay Project SDK contiene:

- Liferay Workspace (proyecto basado en Gradle o Maven para CI y DevOps)
- Nuevos plugins Gradle y Maven que cubren todos los aspectos de los requisitos de creación de un proyecto Liferay, servicios de backend, portlets, frontend y temas modernos de JS.
- Blade CLI (nueva herramienta de línea de comandos para workflows de desarrollo más rápidos)
- Liferay Developer Studio (entorno de desarrollo totalmente integrado y basado en Eclipse)

El Liferay Project SDK cuenta con una simple herramienta para instalar todas las herramientas necesarias para el desarrollo del Liferay DXP. También le dará la opción de configurar su primer Liferay Workspace. Por favor, descargue el instalador adecuado para su sistema operativo (Windows, Linux, OSX).

Liferay Workspace

Una de sus primeras tareas al actualizar su código es migrarlo a nuestra nueva estructura de proyectos basada en Liferay Workspace. Es algo similar a los Plugins SDK originales. Soporta tanto el Gradle como el Maven y proporciona compatibilidad retroactiva para sus proyectos basados en Plugins SDK. También aprovechará todas las nuevas herramientas de temas que creamos en Liferay DXP y se integra con Liferay Developer Studio.

Su nuevo proyecto quedará así:

```
.
├── configs
│   ├── common
│   ├── dev
│   ├── local
│   ├── prod
│   └── uat
├── modules
│   └── contains new OSGi modules
├── themes
│   └── contains new node.js based frontend themes
├── wars
│   └── contains traditional portlet and theme WARs
│       (like those built with PluginsSDK)
├── build.gradle
├── gradle-local.properties
├── gradle.properties
└── settings.gradle
```

Si está viniendo de un proyecto que aprovechó los Plugins SDK, puede mover sus Plugins de SDK como una de las carpetas dentro de Liferay Workspace. Recuerde que el plug-in SDK sólo admite la creación de WARs. No puede crear paquetes configurables OSGi desde los Plugins de SDK.

Blade CLI

Junto con Liferay Workspace, tenemos una nueva herramienta de línea de comandos llamada Blade CLI. Esta herramienta le permite crear aplicaciones, extensiones, etc. como lo haría en un Plugin SDK, pero también proporciona

funcionalidad adicional. Puede iniciar su servidor Liferay o implementar automáticamente su proyecto mientras realiza cambios. Es también la manera de crear un nuevo Liferay Workspace. Si está viniendo de una estructura de diseño de Plugins SDK existente, tenemos una manera de migrar automáticamente a la nueva estructura de proyecto de Liferay Workspace, como se muestra a continuación.

Para crear un nuevo Liferay Workspace:

```
$ blade init workspace-name
$ cd workspace-name
```

Para actualizar un Plugin SDK existente en Liferay Workspace:

```
$ cd plugins-sdk
$ blade init -u
```

Para crear un nuevo módulo:

```
$ blade create -t mvc-portlet module-name
```

Por favor, consulte nuestra documentación para comprobar comandos adicionales proporcionados por Blade.

Principales Cambios

Ahora que su código está actualizado con la nueva estructura de proyecto, tal vez una de las partes más difíciles en actualizar su base de código es, en realidad, saber lo que ha cambiado desde el último release de Liferay y hacer los cambios apropiados en su código. Para el Liferay DXP, tenemos una sección de los principales cambios en nuestra documentación. Se presenta una lista cronológica de cambios que impiden el funcionamiento de funcionalidades existentes, APIs o contratos con desarrolladores externos o usuarios Liferay. Algunos de los tipos de cambio documentados en el documento incluyen:

- Funcionalidades que se han eliminado o reemplazado.
- Incompatibilidades de la API: cambios en las API públicas de Java o JavaScript.
- Cambios en las variables de contexto disponibles para plantillas.
- Cambios en las clases CSS disponibles para los temas y portlets de Liferay.
- Cambios en la configuración: archivos de configuración, como portal.properties, system.properties, etc.
- Requisitos de ejecución: versión Java, versión J2EE, versiones del explorador, etc.
- Descontinuación o fin de soporte: por ejemplo, avisando que un determinado recurso o la API se quitará en una versión futura.
- Recomendaciones: Por ejemplo, recomendamos el uso de una nueva API que sustituye a una API antigua, aunque se mantenga en Liferay para la compatibilidad con versiones anteriores.

Es importante que se familiarice con la lista completa para entender qué cambios puede que tenga que hacer con su base de código. Recomendamos leer la lista de cambios recientes para obtener una idea general de cómo Liferay DXP seguirá evolucionando en futuras versiones.

Para conocer los principales cambios actualmente para Liferay DXP, por favor acceda a esta lista.

Liferay Developer Studio

Liferay Developer Studio es el entorno de desarrollo completo e integrado para Liferay que ya puede utilizar en su proyecto existente. Nuestro Liferay Project SDK puede instalar opcional Liferay Developer Studio. El producto incluye una nueva Code Upgrade Tool para ayudar a actualizar Plugins SDK (o basado en Maven) de su proyecto en 6.X a Liferay Workspace y el Liferay DXP. Incluso si su equipo prefiere otro IDE o entorno de desarrollador, le recomendamos que todavía utilice Liferay Developer Studio al menos para utilizar el Code Upgrade Tool durante el proceso de actualización.

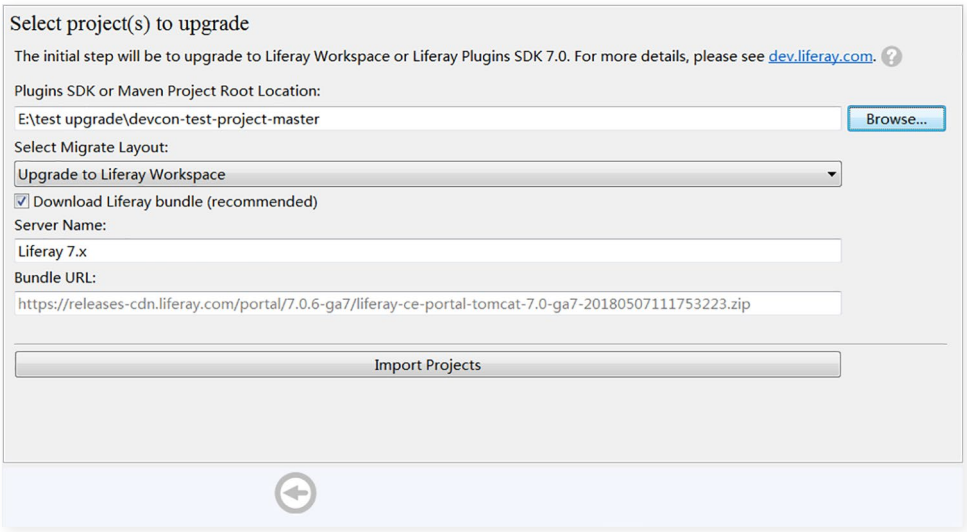
Code Upgrade Tool

El Code Upgrade Tool ofrece los siguientes beneficios:

- Identifica códigos afectados por los cambios de API
- Describe cada cambio de API relacionada al código
- Sugiere como adaptar el código
- Proporciona opciones, en algunos casos, para adaptar el código automáticamente

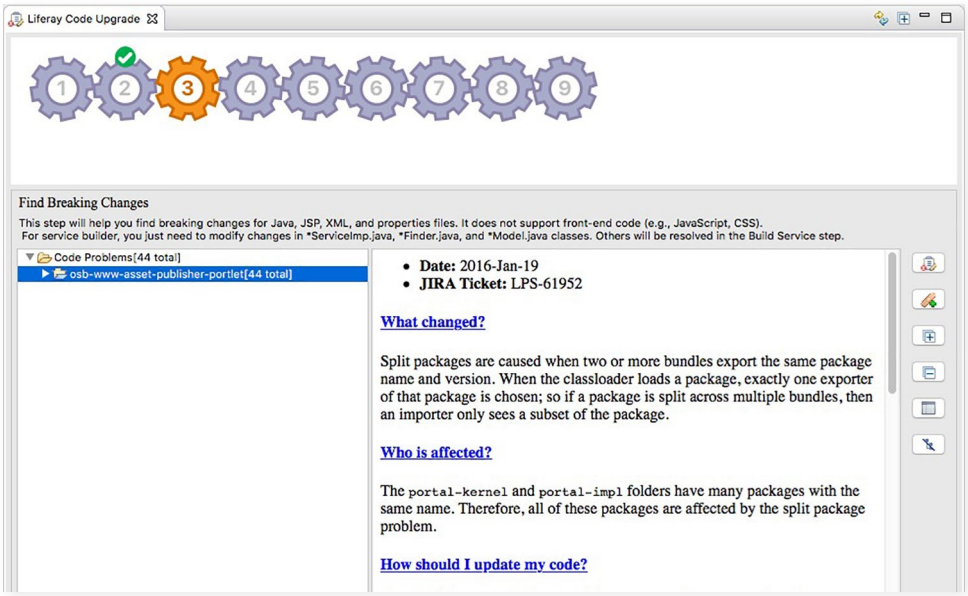
Actualizando los Plugins SDK para Liferay Workspace

Todo lo que usted necesita hacer es demostrar a la herramienta de actualización de código de barras donde sus Plugins SDK de su proyecto en 6.X se encuentran y se pueden actualizar automáticamente a Liferay Workspace.



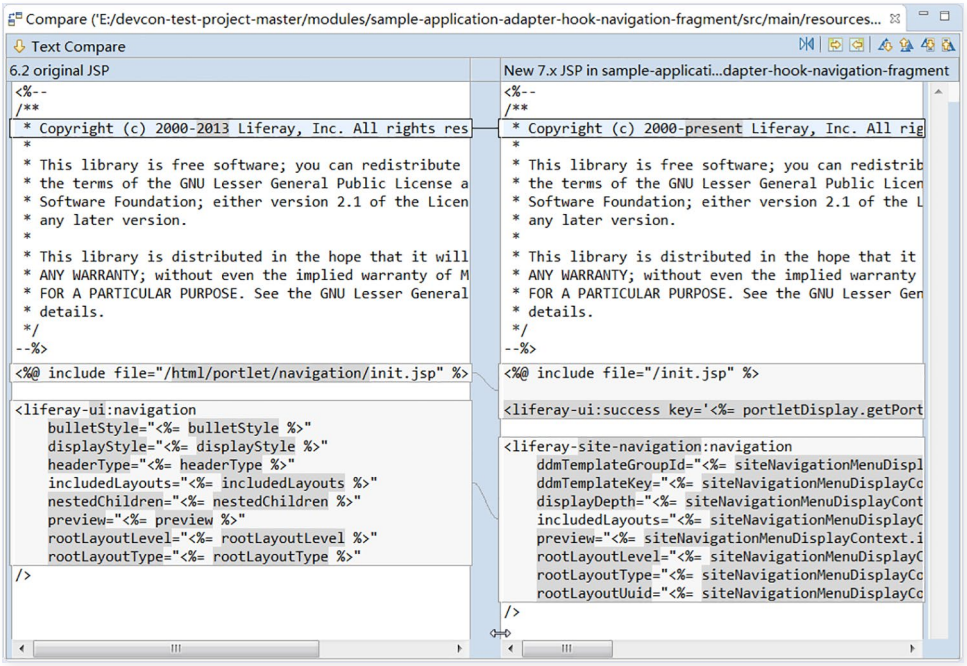
Ubicando los Cambios

Una vez que su código esté en el Code Upgrade Tool, él va a analizar y comparar con la lista de Principales Cambios, ofreciendo una manera sencilla de identificar y tener más detalles sobre estos cambios, además de los puntos de acción necesarios para actualizar su código. En muchos casos, la herramienta puede hacerlo automáticamente para usted.



Actualizando Customizaciones en la Nueva Estructura de Módulos

A veces, no hay como actualizar el código existente y exigir una estructura nueva y más modular, como hooks JSP. El Code Upgrade Tool puede incluso ayudarle en este escenario muy complejo.



Para mayor información sobre como utilizar el Code Upgrade Tool y todas sus funcionalidades, usted puede recurrir a este tutorial detallado.

Escenarios de Actualización de Código

Migrando un 6.2 WAR para un WAR soportado por Liferay DXP

El primer paso que debe realizar es actualizar su portlet 6.2 a un portlet de Liferay DXP. Incluso si usted planea convertir su portlet a módulos OSGi, le recomendamos que actualice su WAR legado para ser soportado por el Liferay DXP primero. Si salta de un módulo heredado de 6.2 WAR para el DXP, será mucho más difícil realizar su depuración y averiguar qué problemas están relacionados con los cambios de la API y cuáles están relacionados con el proceso de migración.

El Upgrade Assistant mencionado arriba encuentra los cambios recientes en su portlet y los actualiza apropiadamente. Asegúrese de actualizar también todas las dependencias de Liferay que ha especificado para el Liferay DXP (por ejemplo, `ivy.xml`, `liferay-plugin-package.properties`, etc.).

Después de completar el proceso de actualización y tener un WAR compatible con Liferay DXP, usted necesita decidir si debe convertir su portlet en un módulo OSGi. Describimos los siguientes escenarios que le ayudarán a tomar su decisión.

Convirtiendo un portlet en un módulo OSGI

Ahora que su workspace se ha creado y sus portlets están actualizados para el Liferay DXP con soporte WAR, podemos considerar la conversión de su portlet a un módulo OSGi.

USTED DEBE CONVERTIR CUANDO:

- Usted tiene aplicaciones muy grandes con muchas líneas de código. Por ejemplo, si hay muchos desarrolladores que colaboran en una aplicación simultáneamente y haciendo cambios con frecuencia, separar el código en módulos aumentará la productividad y proporcionará la agilidad para liberar con más frecuencia.
- Su plugin tiene partes reutilizables que usted quisiera consumir en otro lugar. Por ejemplo, suponga que tiene una lógica de negocio que se reutiliza en varios proyectos diferentes. En lugar de copiar este código en varias diferentes WAR e implementar estos WAR en diferentes lugares, puede convertir su aplicación en módulos y consumir los servicios proporcionados por estos módulos en otros módulos.

USTED NO DEBE CONVERTIR CUANDO:

- Usted tiene un portlet compatible con JSR-168/286 y todavía desea implementarlo en otro contenedor de portlet. Si desea mantener esta compatibilidad, se recomienda continuar con el modelo WAR tradicional.
- Usted está utilizando un complejo marco web heredado que está fuertemente vinculado al modelo de programación Java EE, y la cantidad de trabajo necesario para hacer funcionar con OSGi es mayor de lo que usted necesita.
- Su plugin interactúa con otras características del servidor de aplicaciones JEE, por ejemplo, EJBs, message driven beans, etc. Las aplicaciones basadas en módulos no son tan portátiles cuando interactúan directamente con el servidor de aplicaciones.
- La intención original de sus aplicaciones legadas es tener una duración limitada.

Si decide convertir su portlet a un módulo OSGi, lo guiaremos a continuación.

Para un portlet: `blade create -t mvc-portlet [APPLICATION_NAME]`

Si usted necesita de un Service Builder: `blade create -t servicebuilder -p [ROOT_ PACKAGE] [APPLICATION_NAME]`

La primera cosa que usted se dará cuenta es que sus proyectos ahora utilizan la estructura de diseño Maven estándar.

```
.
├── bnd.bnd
├── build.gradle
└── src
    └── main
        ├── java
        │   └── com
        │       └── liferay
        │           └── samples
        │               └── servicebuilder
        │                   └── web
        │                       └── JSPPortlet.java
        └── resources
            ├── META-INF
            │   └── resources
            │       ├── css
            │       │   ├── _partial.scss
            │       │   └── main.scss
            │       ├── edit_foo.jsp
            │       ├── foo_action.jsp
            │       ├── icon.png
            │       ├── init.jsp
            │       └── view.jsp
            └── content
                └── Language.properties
```

En su workspace, el archivo bnd.bnd es muy importante, ya que aplicará automáticamente el plugin liferay-gradle a su proyecto Gradle. El plugin liferay-gradle aplicará el plugin Gradle Java junto con otros plugins de Liferay que son muy útiles, como css-builder, source-formatter y lang-builder.

Usted notará que usted no necesita un archivo portlet.xml / liferay-portlet.xml. El contenido del archivo se debe transferir a la clase de portlet de Java.

```
@Component(
    immediate = true,
    property = {
        “com.liferay.portlet.display-category=category.sample”,
```

```
            “com.liferay.portlet.icon=/icon.png”,
            “javax.portlet.name=1”,
            “javax.portlet.display-name=Tasks Portlet”,
            “javax.portlet.security-role-ref=administrator,guest,power-user”,
            “javax.portlet.init-param.clear-request-parameters=true”,
            “javax.portlet.init-param.view-template=/view.jsp”,
            “javax.portlet.expiration-cache=0”,
            “javax.portlet.supports.mime-type=text/html”,
            “javax.portlet.resource-bundle=content.Language”,
            “javax.portlet.info.title=Tasks Portlet”,
            “javax.portlet.info.short-title=Tasks”,
            “javax.portlet.info.keywords=Tasks”,
        },
        service = Portlet.class
    )
    public class TasksPortlet extends MVCPortlet {
```

Si ha creado una plantilla para la construcción de servicios, se dará cuenta de que se han generado dos plugins diferentes:

`plugin-name-api` - Aquí es donde las interfaces de sus servicios quedarán.

`plugin-name-service` - Aquí es donde las implementaciones de sus servicios se quedarán. El paquete será similar al portlet arriba.

Actualizando Temas

En el Liferay DXP, creamos un nuevo conjunto de herramientas de temas con los que los desarrolladores de frontend deben estar más familiarizados. Se construyen usando el Node.js, yo y gulp. Si tiene un tema existente en su Plugins SDK, puede migrarlos al nuevo Liferay Workspace.

Las nuevas herramientas de tema también ayudan a facilitar el proceso de actualización del tema. Se actualiza de 6.x a 7.0 / 7.1 o 7.0 a 7.1, usted seguiría utilizando.

```
gulp upgrade
```

Sin embargo, hay algunas diferencias entre realizar la actualización de 6.1 a 7.0 y 6.2 a 7.0. Por favor, consulte la documentación apropiada para estas distinciones:

1. Upgrading to DXP 7.2
2. Upgrading from Liferay Portal 6.1 to 7.0

Al realizar la actualización de 7.0 a 7.1, muchos de sus temas del DXP 7 deben seguir funcionando. Es posible que desee ejecutar las herramientas de actualización de temas. Puede encontrar más detalles en el Centro de ayuda (Customer Portal).

Desde la actualización a Bootstrap 4 de DXP 7.1, hay descontinuación y eliminación en el estilo CSS que se deben ejecutar. La lista de cambios de CSS se puede encontrar en la documentación oficial del DXP. Aquí puede encontrar una guía para actualizar su tema de 6.2, 7.0 y 7.1 a 7.2.

El cambio final en los temas es la eliminación de las plantillas Velocity. Las plantillas Velocity se interrumpieron en el DXP 7.0 dando lugar a las plantillas Freemarker. Si su proyecto en DXP 7.0 continuó usando Velocity, usted debe convertirlos al Freemarker.

Recursos Adicionales

- Liferay DXP Upgrade Reference Guide — Lea antes de planificar o ejecutar su upgrade.
- Liferay Developer Guide — Proporciona tutoriales importantes para varias etapas del proceso de actualización de código. Las siguientes secciones son extremadamente importantes para revisar y entender, especialmente al decidir cuándo convertir complementos basados en WAR en módulos OSGi.
    - What's Changed and What Hasn't
    - Migrating from Plugins SDK to Liferay Workspace
    - Planning Upgrades and Optimizations for WAR based plugins
    - Upgrading Plugins
        - Upgrading Themes from 6.x to 7.0
        - Upgrading Themes from 7.0 to 7.1
    - Upgrading and Migrating From EXT
- Liferay Developer Tools
    - Liferay Workspace — Sustitución de los Liferay Plugins SDK
    - Blade CLI — Herramienta de línea de comando para desarrollar en Liferay DXP
    - Liferay Developer Studio
    - Liferay Workspace Maven Edition — Una serie de puntos de integraciones Maven para desarrollar con Liferay
- Liferay Blade Samples — Repositorio de ejemplos de códigos y casos de uso.

Resumen

Para aquellos que optan por seguir adelante, una actualización para el Liferay DXP tiene el potencial de desbloquear toda la gama de beneficios de la versión más reciente de la plataforma Liferay. Pero cada empresa debe determinar por sí misma si una actualización será beneficiosa para la dirección del negocio, después de pesar cuidadosamente los costos, riesgos, plazos, mano de obra y beneficios comerciales involucrados.

Siguiendo Adelante

Nuestro equipo de Global Services está disponible para proporcionar un análisis profundo de sus requerimientos específicos, crear un plan de actualización personalizado y ofrecer una experiencia para preparar su empresa para el éxito con Liferay. Más información sobre el Liferay Digital Experience Platform y los servicios de consultoría disponibles entrando en contacto con sales-latam@liferay.com.

Liferay desarrolla software que ayuda a las organizaciones a crear experiencias digitales en la web, el móvil y en todo tipo de dispositivos conectados. Nuestra plataforma es open source, lo que hace posible una mayor fiabilidad, innovación y seguridad. Nuestra empresa intenta dejar una impronta positiva en el mundo, a través del negocio y de la tecnología. Cientos de organizaciones del sector financiero, salud, gobierno, asegurador, retail, industria, y otros múltiples mercados utilizan Liferay. Visítenos en liferay.com

200122