

The background image is a composite. It features a woman on the left and a man on the right, both with purple markings on their faces. A hand is visible in the foreground, holding a pen and pointing towards a tablet. The overall color scheme is a mix of purple and blue.

Merz Aesthetics Click

Manual de aplicación

Contenido

Merz Aesthetics Click

Manual de aplicación

Tabla de Contenido

_Estilos de pantalla

Funcionamiento de la aplicación3

Diagrama de secuencia de la comunicación con el servidor4

Requerimientos de Asset Bundles de Contenido5

 Técnicas5

 Productos5

 Composición de Prefab6

 Lenguajes6

 Barra de navegación7

 Proceso de creación de asset bundles7

 ItemContent8

 TechCard9

 ProductSmall10

 ProductBig10

 Logo10

 ProductsBackground11

 TechniquesBackground11

Funcionamiento de la aplicación

Funcionamiento de la aplicación

La aplicación en Unity consta de cuatro escenas.

SplashScene: Escena que contiene código de inicialización de realidad aumentada además de que aquí se verifica si el usuario ha hecho log in antes y de ser así descargar imágenes de noticias, productos y técnicas si es que hay una conexión a internet disponible. De no ser así, carga contenido previamente almacenado.

LoginScene: Escena en la que el usuario puede entrar a la aplicación usando correo y contraseña además de pedir un cambio de contraseña. Cuando el usuario loguea, la aplicación guarda toda la información obtenida para poder usarse sin conexión en una sesión futura.

HubScene: Escena donde el usuario puede ver **noticias, productos, técnicas, tutoriales e información del usuario**. Toda la información necesaria para poblar esta escena viene cargada desde la escena de splash o de login excepto por el contenido de los asset bundles que son cargados y almacenados únicamente cuando el usuario hace click para visualizar el contenido. Una vez descargados los contenidos se almacenan con una versión que es verificada cada vez que se obtiene la información del servidor. Si la versión obtenida por el servidor es mayor a la almacenada la aplicación intentará descargar los contenidos nuevamente. De esta manera se pueden actualizar contenidos sin necesidad alguna de recompilar y redistribuir la aplicación.

ShowcaseScene: Escena donde se muestra el contenido de los asset bundles. Este contenido es una secuencia lineal de pasos y puede ser en un ambiente virtual o un ambiente de realidad aumentada. Esto depende de la configuración del asset bundle.

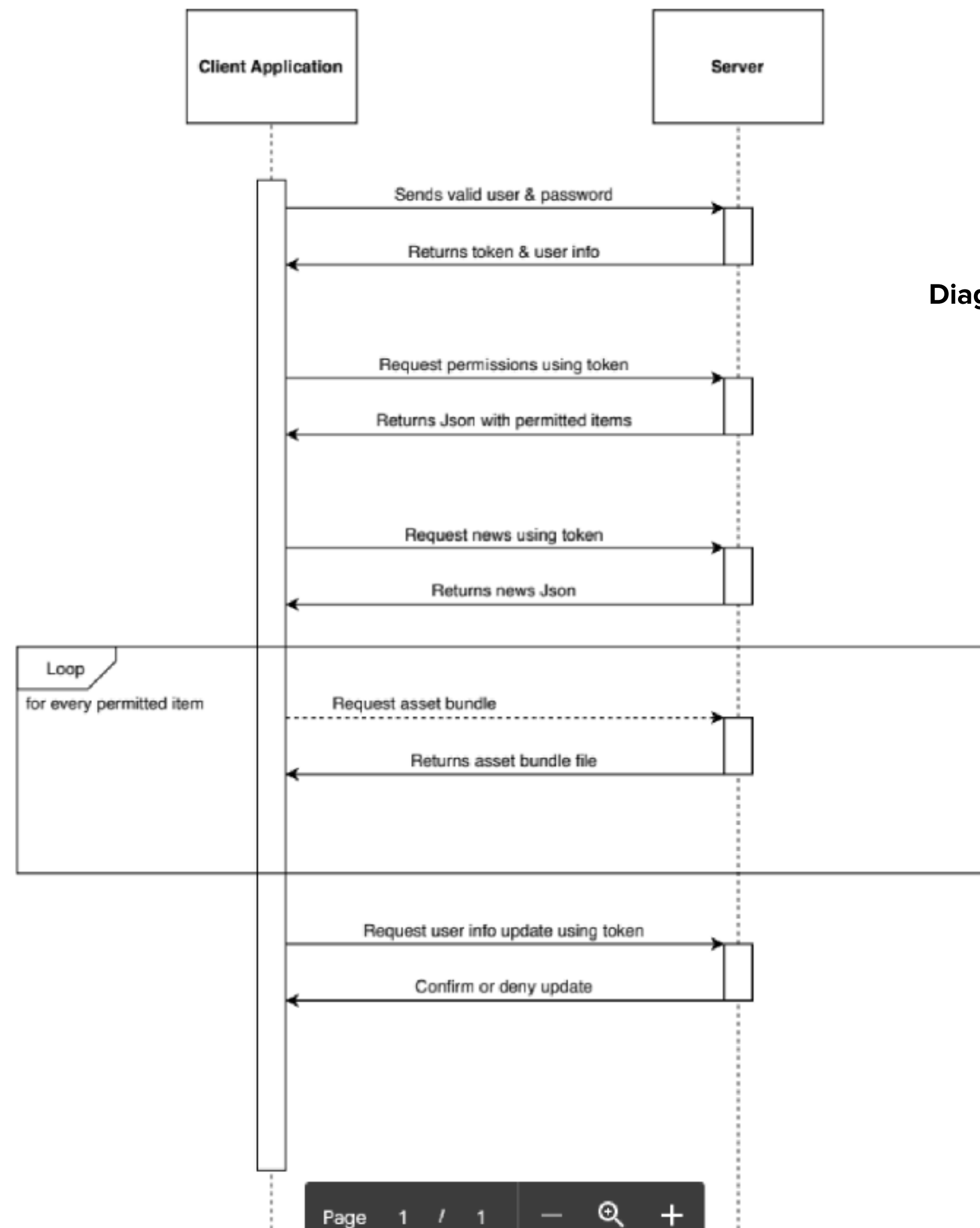


Diagrama de secuencia de la comunicación con el servidor

Nomenclatura

Los asset bundles necesitan llevar una nomenclatura específica para que la aplicación pueda extraer y organizar la información de manera correcta. Esta nomenclatura consta de tres partes separadas por guión bajo. Las tres partes son: nombre de marca, indicador de técnica o producto y finalmente nombre de la técnica o producto.

Ejemplos:

- **belotero_p_volume**
- **belotero_t_volumeintense**
- **belotero_t_bup**

Visualización

Técnicas

Para la visualización de una técnica son necesarios 4 archivos prefab:

- **ItemShowcase_M_AR**
- **ItemShowcase_M_3D**
- **ItemShowcase_F_AR**
- **ItemShowcase_F_3D**

Estos son necesarios para sustentar 2 de los parámetros de la visualización que resultan en 4 diferentes escenarios. Una visualización 3D o de realidad aumentada, con un modelo de hombre o mujer.

Productos

Para la visualización de un producto son necesarios 2 archivos prefab:

- **ItemShowcase_AR**
- **ItemShowcase_3D**

Estos archivos de prefab siguen las mismas reglas que cualquier asset bundle. Cualquier referencia a cualquier asset, sea mesh, material, textura, sprite, sonido, animación, archivo de texto y cualquier otra dependencia debe estar contenido dentro del mismo bundle para que Unity pueda instanciarlos y resolver las dependencias en runtime. La aplicación se encargará de cargar todo dicho contenido.

Requerimientos de Asset Bundles de Contenido

Composición de Prefab

El objeto raíz del prefab necesita contener un script de tipo **ItemShowcase**. A su vez los objetos hijos del prefab que vayan a formar parte de los pasos de la visualización deben contener un script tipo **ShowcasePage**, y deben estar asignados en la lista del ItemShowcase, para que el controlador de la escena pueda “recorrer” estos game objects apagando y prendiéndolos según sea necesario. El prefab puede contener otros objetos que no formen parte de la secuencia de visualización y que necesiten estar activos en todo momento como luces, sonidos, componentes de navegación, etc.

Lenguajes

Para agregar lenguajes a una visualización es necesario que dentro del bundle exista al menos un archivo **json** llamado **loc_** seguido del sufijo del lenguaje e.g. **loc_es**, **loc_en**, **loc_pr**. El arreglo de lenguajes dentro de **ItemContent** es el que determinará en la interfaz qué banderas son visibles para escoger idioma y a la hora de cargar la visualización se buscará el archivo **loc_** correspondiente. En este momento solo están soportados español (es), inglés (en) y portugués (pr). **ItemContent** se configura dentro del servidor en la sección de assets.

El archivo de lenguaje **loc_xx** lleva la siguiente estructura:

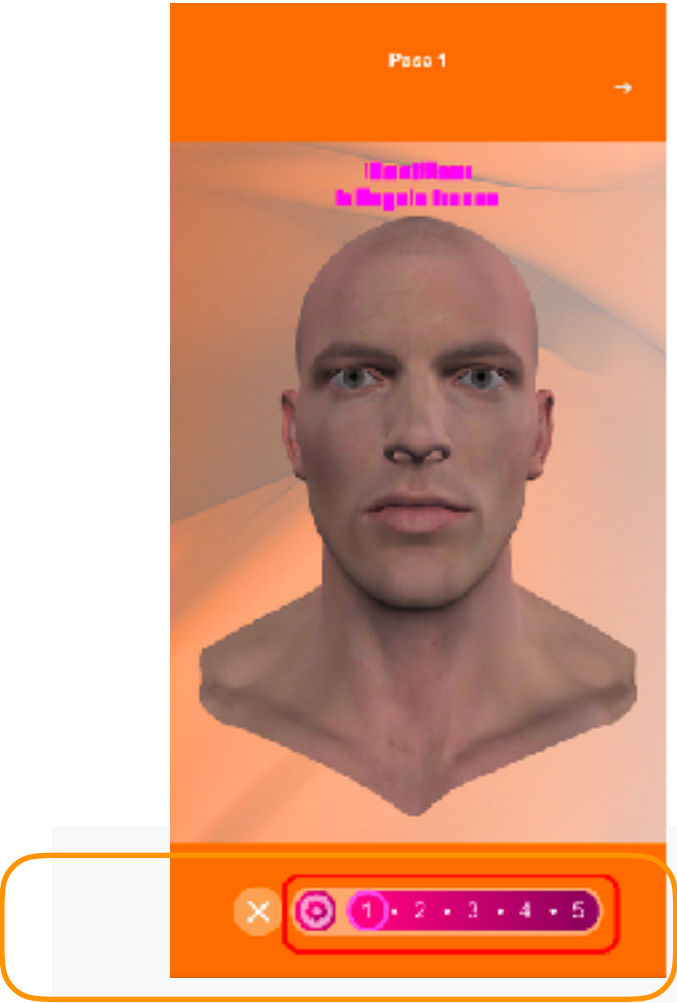
```
{
  "titles": [
    "Title 1",
    "Title 2",
    ""
  ],
  "content": {
    "ref" : "references"
    "any_key" : "any content"
  }
}
```

El arreglo de **titles** contiene todos los títulos que se utilizan en el header de **ShowcaseScene**. El último título se deja vacío porque generalmente el último paso es para referencias y no lleva título pero es opcional. El objeto de **content** es un diccionario donde se puede incluir cualquier string de esta manera un script puede tener asignada una llave que se espera estar dentro de este diccionario y así obtener el contenido. El único campo obligatorio dentro de este diccionario es la **“ref”** que es el texto que se utiliza para poblar el apartado de referencias en la UI.

Requerimientos de Asset Bundles de Contenido

Barra de navegación

La barra de navegación es un prefab opcional de UI para agregar a la parte del footer de una visualización. Sí la aplicación encuentra un prefab con el nombre **NavigationBar** dentro del asset bundle, la aplicación intentará instanciarlo y colocarlo en medio de la navegación. Este prefab de navegación deberá llevar asignado el script de **NavigationBar** para poder ser usado por la aplicación



Proceso de creación de asset bundles

Una vez que los asset bundles contienen todo lo necesario para su correcto funcionamiento es necesario exportarlos para las plataformas de **Android** y **iOS**. Una vez exportados y teniendo la nomenclatura correcta es necesario subir estos archivos al servidor por medio del dashboard en la sección de administración de assets, para que la aplicación pueda los contenidos

Requerimientos de Asset Bundles de Contenido

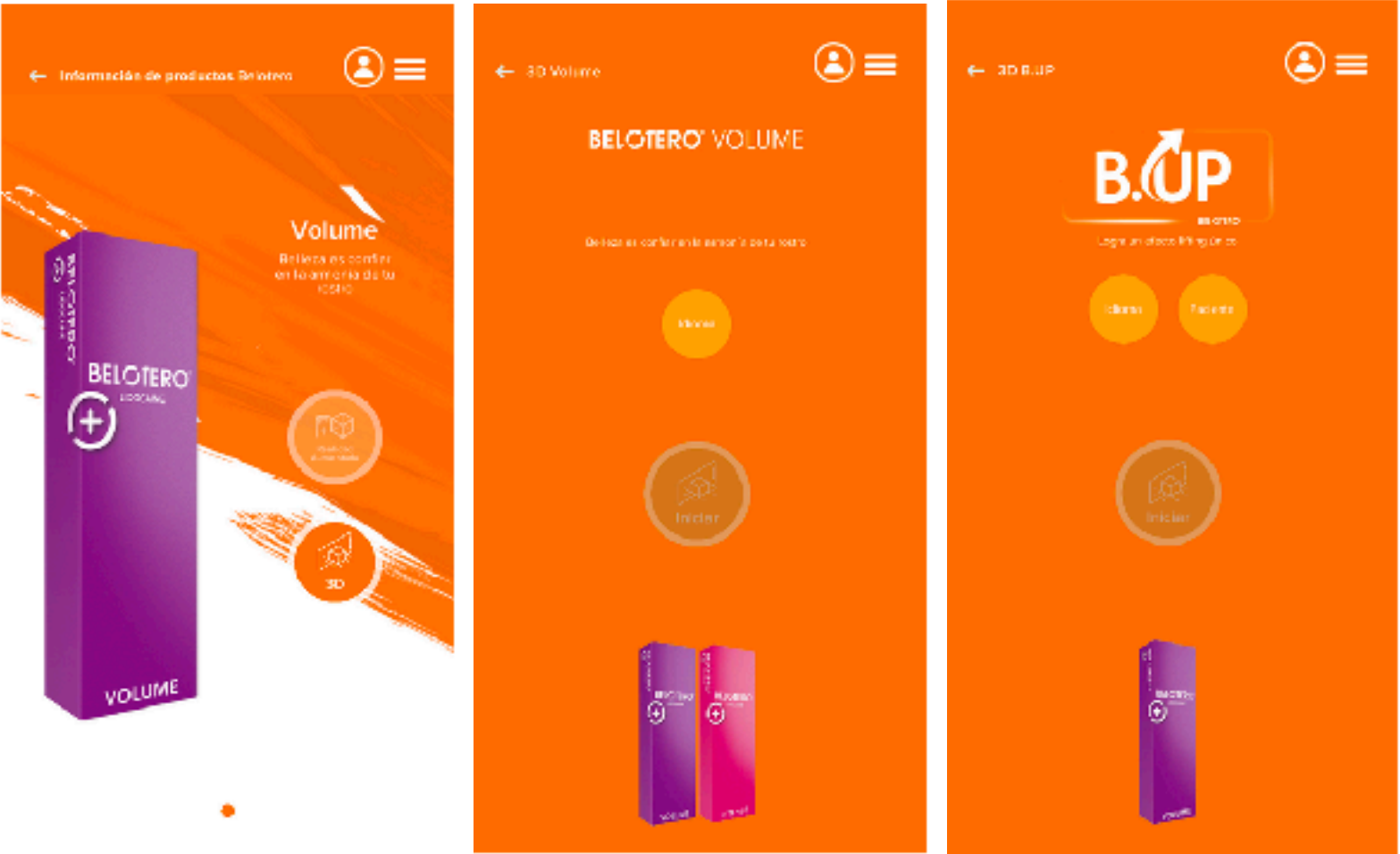
Pre visualización

ItemContent

String **json** que proviene del servidor y que tiene la siguiente estructura:

```
{
  "prettyName" : "Volume",
  "slogan" : "Belleza es confiar en la armonía de tu rostro",
  "languages" : ["es","en","pr"]
}
```

El campo **prettyName** se utiliza en el header del hub. El **slogan** se utiliza en la pantalla de productos, inicio de técnica e inicio de producto.



El campo de **languages** como se mencionó anteriormente se utiliza para saber qué archivos tipo **loc_** se podrán encontrar dentro de dado asset bundle.

Requerimientos de Asset Bundles de Contenido

Imágenes importantes

Estas imágenes deben de ser subidas al servidor a través del dashboard para que la aplicación pueda descargarlas y usarlas en los lugares apropiados. Para más detalles de cómo subir estas imágenes es necesario consultar la documentación del servidor.

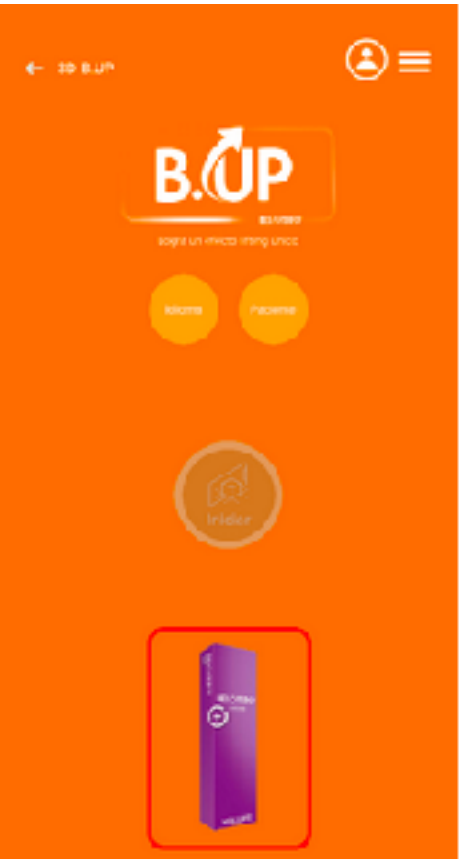
TechCard

Imagen **.png** con tipo de textura en unity como **Sprite (2D and UI)** con medidas de máximo **653 x 313**. Después de esas dimensiones la imagen puede distorsionarse en la aplicación. Se utiliza en la pantalla de técnicas.



ProductSmall

Imagen **.png** con tipo de textura en unity como **Sprite (2D and UI)** con medidas de máximo **300x300**. Después de esas dimensiones la imagen puede distorsionarse en la aplicación. Se utiliza en las pantallas de inicio de producto y técnica.



Requerimientos de Asset Bundles de Contenido

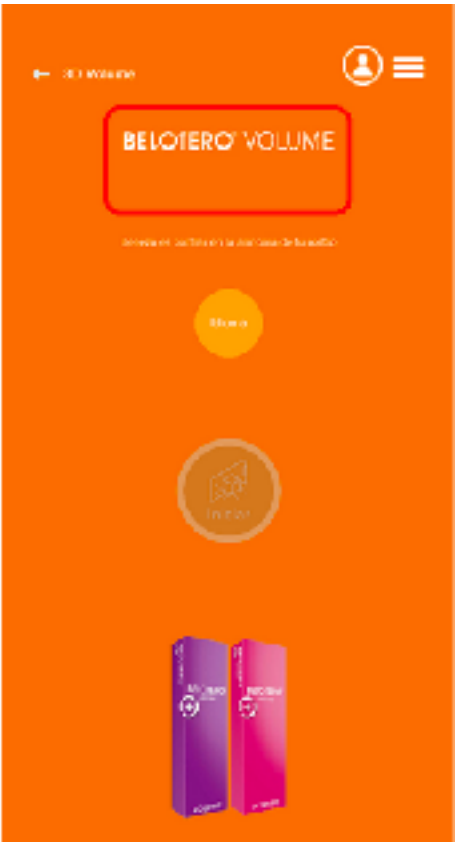
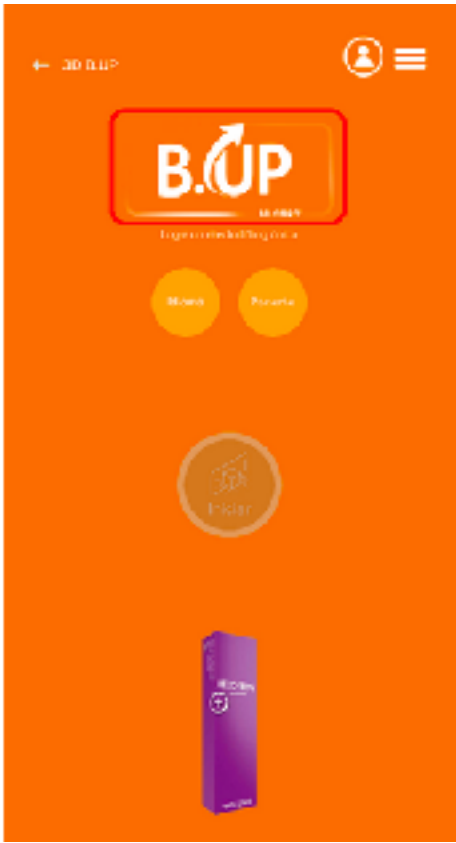
ProductBig

Imagen .png con tipo de textura en unity como **Sprite (2D and UI)** con medidas de máximo **300x800**. Después de esas dimensiones la imagen puede distorsionarse en la aplicación. Se utiliza en la pantalla de productos.



Logo

Imagen .png con tipo de textura en unity como **Sprite (2D and UI)** con medidas de máximo **347 x 156**. Después de esas dimensiones la imagen puede distorsionarse en la aplicación. Se utiliza en las pantallas de inicio de producto y técnica



Requerimientos de Asset Bundles de Contenido

ProductsBackground

Imagen .png con tipo de textura en unity como **Sprite (2D and UI)** . Se utiliza como fondo en la pantalla de productos. Es necesario que todos los asset bundles de la misma marca tengan asignados la misma imagen de background de producto.



TechniquesBackground

Imagen .png con tipo de textura en unity como **Sprite (2D and UI)** . Se utiliza como fondo en la pantalla de técnicas. Es necesario que todos los asset bundles de la misma marca tengan asignados la misma imagen de background de técnica.

