

Arbolado – Restricciones para funcionamiento en modo Off-line

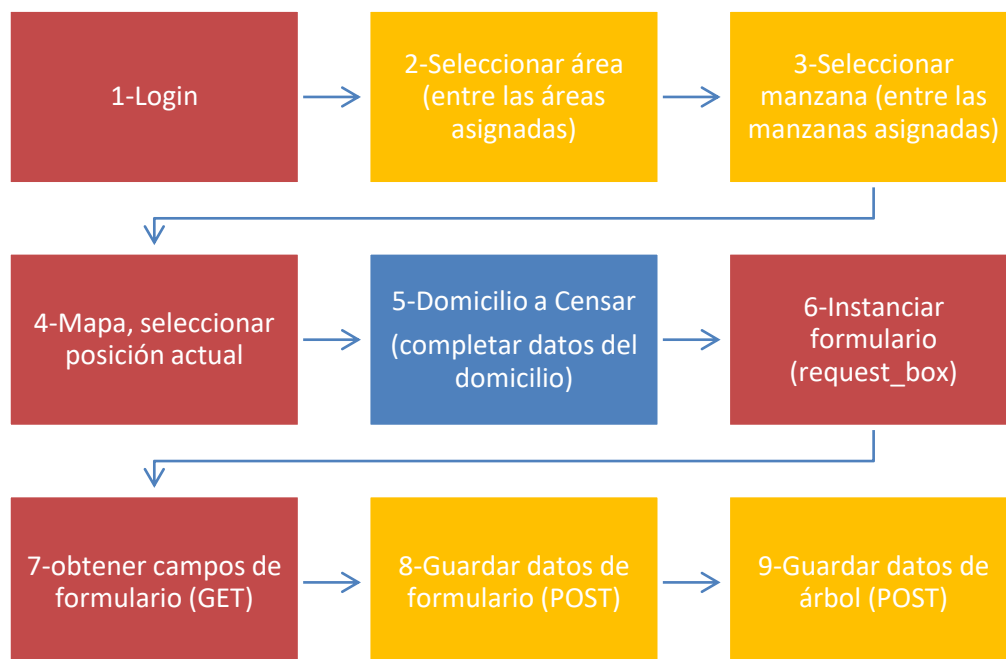
El presente documento tiene como objetivo exponer las restricciones actuales que posee el sistema (front-end) Arbolado para funcionar en modo off-line (sin conexión a la red), detallar la estrategia para modificar las restricciones “salvables” y justificar las que no pueden ser evadidas.

Antes de continuar con el desarrollo del documento, es de suma importancia destacar que la arquitectura de solución y el diseño técnico impuestos inicialmente suponen un alto nivel de dependencia entre el front end y el back end ya que todas las pantallas de la aplicación requieren del consumo del API rest del backEnd. Esto, lejos de facilitar el funcionamiento en modo off-line, implica la implementación de una serie de mecanismos de contingencia para poder implementar dicho funcionamiento.

A modo ilustrativo, se present un diagrama que muestra las actividades dentro del sistema (alto nivel). En rojo, las actividades que se ven afectadas actualmente si el usuario no posee conexión a la red.

En amarillo las actividades que si bien tienen un fuerte nivel de dependencia el backEnd, podría incluirse algún tipo de mecanismo de salvaguarda para que el usuario pueda avanzar en esa actividad aún cuando no posea conectividad.

El siguiente diagrama corresponde con el censo de un nuevo árbol.



- 1) Para realizar el login (validación de credenciales correctas y obtención de token de sesión) es necesario tener conectividad contra el backEnd.
- 2) Sobre este punto se estableció un mecanismo de obtención y almacenamiento (en local storage) de los datos devueltos por el servicio "TreeList". A partir de esto, es posible navegar las pantallas de áreas asignadas y manzanas asignadas. Sin embargo, se requiere que al menos el usuario inicie sesión con conectividad para poder obtener los datos una primera vez. Por lo tanto esto no es 100% offline
- 3) Mismas consideraciones que en el punto 2.
- 4) La pantalla que muestra la ubicación actual del censista utiliza el componente de mapas de Google Maps. Este componente, para renderizarse, valida un API KEY. Esta validación se hace en línea y ANTES de renderizar el componente en pantalla, por lo tanto el mismo no puede ser instanciado si antes no se establece que la API Key es correcta y válida. Dicho esto, la pantalla en la que el usuario selecciona su ubicación y luego hace click en el botón siguiente, no se renderizaría (a nivel de dom) de acuerdo a lo esperado, cosa que afecta el normal funcionamiento de la aplicación. Hemos investigado si google maps posee de algún tipo de mecanismo en su API para funcionar en modo offline habiendo encontrado que esta funcionalidad está en roadmap pero sin fecha cierta aún
<https://accounts.google.com/ServiceLogin/webreauth?passive=1209600&continue=https%3A%2F%2Fissuetracker.google.com%2Fissues%2F35827808&followup=https%3A%2F%2Fissuetracker.google.com%2Fissues%2F35827808&authuser=0&flowName=GlifWebSignIn&flowEntry=ServiceLogin>. Más allá de la validación del API Key, también está la imposibilidad de que se descargue toda la información necesaria para poder "pintar" en pantalla el fragmento de mapa necesario (nombres de calles, capas con las manzanas, direcciones, etc, etc).

Hay otro inconveniente en relación a la obtención de este dato que es el siguiente: Si bien muchos dispositivos cuentan con antena GPS, el posicionamiento nativo del navegador, utiliza el posicionamiento por triangulación utilizando las antenas de la red celular, con lo cual, si el dispositivo no cuenta con conexión a la red, el problema deja de ser solamente un tema de mostrar un mapa en pantalla para convertirse en una imposibilidad absoluta de obtener la posición actual del censista.

Dado que este es un dato crítico y absolutamente obligatorio para la aplicación, se hace imposible continuar con el censo tal como se ha establecido en el análisis funcional.

- 5) Salvando los puntos 1 a 4, esta pantalla no presentaría problemas para funcionar fuera de línea.
- 6) Llegando a la pantalla de censado en sí misma, lo primero que se hace es crear una instancia de formulario, cosa que debe hacerse mediante una llamada a servicio rest del lado del backEnd. Esta creación de instancia de formulario es MANDATORIA para poder llamar al servicio que devuelve los campos de formulario a pintar en la pantalla ya que para invocar al mismo hay que pasar como parámetro, el ID de instancia de formulario. Este paso no puede realizarse sin conectividad al backEnd
- 7) Este paso involucra la llamada al servicio que devuelve todos los elementos del formulario, datos que son utilizados para poder renderizar en pantalla los diferentes

campos que el usuario debe completar para llevar adelante el censo. Esto no puede realizarse en modo off-line.

Una aclaración es que el paso 7 depende de 6 y deben ser ejecutados SIEMPRE de manera secuencial, por lo tanto, si bien podría pensarse en algún tipo de mecanismo de procesamiento batch de estas llamadas, las mismas se realizarían cuando el usuario recupere conexión a la red, con lo cual, no tendría ya ningún tipo de aporte de valor. Esta implementación es compleja ya que requiere de un manejo de colas de transacciones, control de excepciones y manejo de errores que no es trivial. Adicionalmente estaríamos logrando solamente que se creen las instancias de formulario, pero sin datos, por lo tanto, el usuario luego tendría una serie de formularios vacíos e instanciados que no le aportarían valor en absoluto.

8 y 9)

El almacenamiento de datos también requieren conectividad con el backEnd. Y dado que no es posible (por lo expuesto en 6 y 7) pre-guardar los formularios completos, tampoco sería viable desde lo técnico crear una cola de transacciones para los puntos 8 y 9.

Cabe destacar que por cada árbol censado, es necesario guardar, por un lado la información del formulario y por otro lado la información del árbol (que incluye el ID del formulario correspondiente).

En relación al circuito de modificación de árboles ya censados, la situación es similar, con algunas diferencias mínimas. Por ejemplo, en la pantalla de formulario no es necesario invocar al servicio request_box, pero al mismo tiempo, al no poderse invocar en modo off-line al servicio que devuelve un formulario, sería inviable (por ejemplo) pintar los campos en la pantalla de censo. Tampoco sería posible pintar el mapa que permite seleccionar un árbol determinado por lo que no sería posible avanzar en el escenario funcional.