MANUAL TÉCNICO

Sistema Web Transur

Introducción

El manual técnico, es la guía práctica para la actualización de archivos y la publicación de la aplicación web Transur.

Al tener JavaScript del lado del cliente y del lado del servidor, es indispensable poseer conocimientos intermedios de este lenguaje de programación y administración de Linux, tal como se citó en las conclusiones y recomendaciones.

El contenido del presente manual técnico es conciso y dirigido para programadores de la pila de aplicaciones MEAN.IO

Software para la aplicación Transur

Antes de empezar con el manual técnico se debe de instalar el siguiente software (Incluido en el cd)

Plataforma Windows

1. Git-1.9.5-preview
2. Mongodb-win32-i386-2.2.7
3. Node-v0.10.20-x86.msi
4. JetBrains WebStorm v8.0.3

Las siguientes versiones variaran con relación al hardware y al sistema operativo utilizado:

Git es utilizado como una consola Linux, que permitirá instalar un directorio local donde se insertaran los archivos de la aplicación

Mongodb, es una base de datos no relacional, que se acopla perfectamente con Node.js

Node.js es un entorno de ejecución para utilizar JavaScript del lado del servidor

JetBrains Web Storm v 8.0.3, es un IDE de desarrollo amigable con los programadores de Node.js

Creación de un nuevo proyecto local

Al abrir Git, con el comando cd ingresamos a una carpeta creada previamente caso contrario la creamos con el comando mkdir, como en Fig. 1.

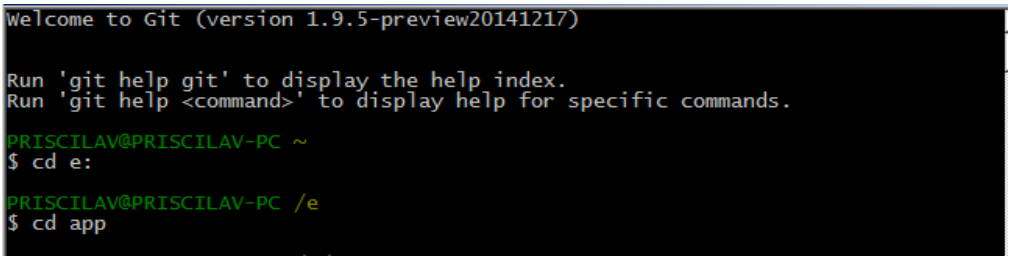


Fig. 1 Directorio de la aplicación

Una vez que se ingresó al directorio se tipea el comando **$ npm install -g gulp**, que instala un automatizador de tareas y el comando: **$ npm install -g mean-cli** encargado de instalar las herramientas del cliente MEAN.IO

Fig. 2 Instalación del automatizador de tareas y herramientas del cliente MEAN.IO

Ahora se instalan los archivos del directorio local con el comando: ***mean init nombredelaplicacion,*** una vez ejecutado se muestra en consola los directorios de la aplicación local ilustrados en la Fig. 3:

```
PRISCILAV@PRISCILAV-PC /e/app
$ cd elapp

PRISCILAV@PRISCILAV-PC /e/app/elapp (master)
$ ls
LICENSE     bower.json        gulp          mean.json     packages
Procfile    bower_components  gulpfile.js   node_modules  server.js
README.md   config            karma.conf.js package.json  tools
```

Fig. 3 Archivos de la aplicación

Ejecutamos webStorm 8 y tendremos el listado de directorios del proyecto de la Fig. 4

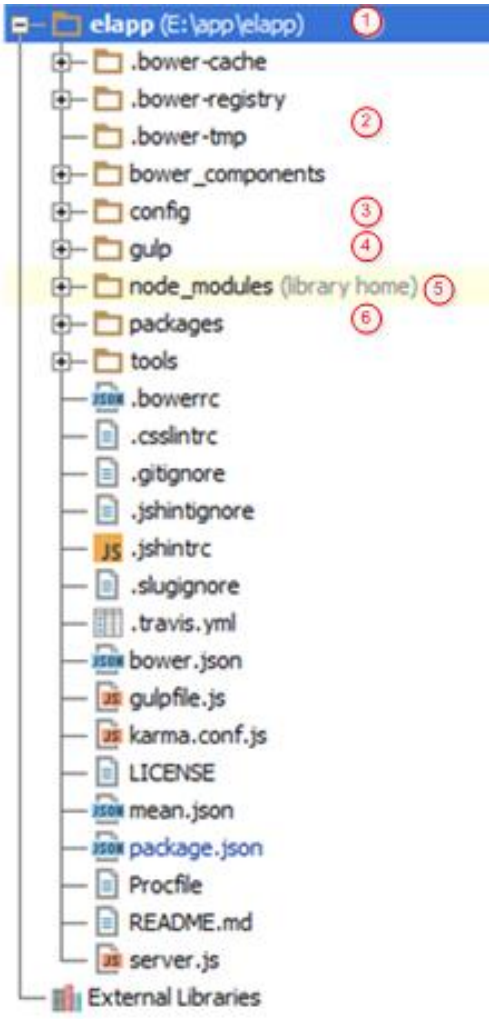


Fig. 4 Directorios de la aplicación

El directorio raíz **/elapp (1)**, contiene todos los archivos de la aplicación. Las carpetas cuyo nombre empieza por **/bower (2)**, son el contenido del gestor de paquetes para el front-end, donde residen principalmente los pakages de angular y librerías utilizadas por la aplicación, como las librerías encargadas de las exportaciones de los reportes de Transur.

La carpeta **/config (3),** posee los archivos de configuración (**assets.json**) de la aplicación para que la misma pueda ser ejecutada desde el entorno local como en el servidor.

El archivo **assets.json**, posee una estructura tipo json, en el cual se deben de citar los archivos: js, css y archivos extras de la aplicación como el archivo FileSaver.js para la exportación de archivos. Los archivos listados serán cargados en el despliegue de cada uno de los htmls.

La carpeta **/gulp (4)**, tiene el contenido del paquete encargado de automatizar las tareas de ejecución y enlace de los archivos. js y css.

La carpeta **/node_modules (5)**, contiene las librerías angular, bower de la aplicación con la pila MEAN.IO

La carpeta **/packages (6),** contiene los packages de la aplicación, contienen 3 directorios **/core, /custom** y **/contrib,** el directorio /core contiene los packages theme y users los cuales administran los temas de presentación de la aplicación y los usuarios del sistema, los paquetes creados por el programador se alojan en el directorio **/custom** y **/contrib** aloja los paquetes creados por los desarrolladores de la pila de aplicaciones MEAN.IO.

El directorio **/custom,** es un directorio que depende principalmente de la versión o actualización de MEAN.IO, en algunas versiones no existe pero es necesario tener en cuenta que de existir los packages creados por el programador residirán en este directorio.

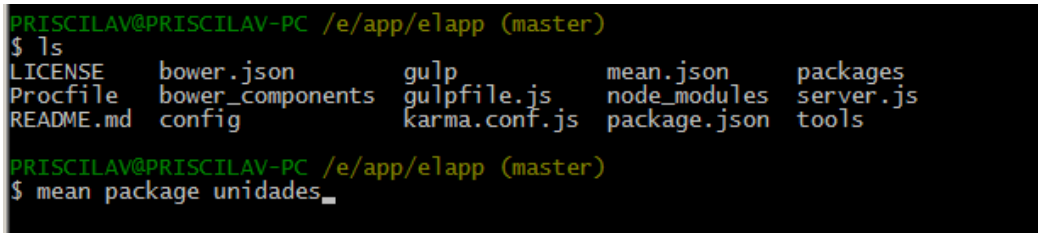
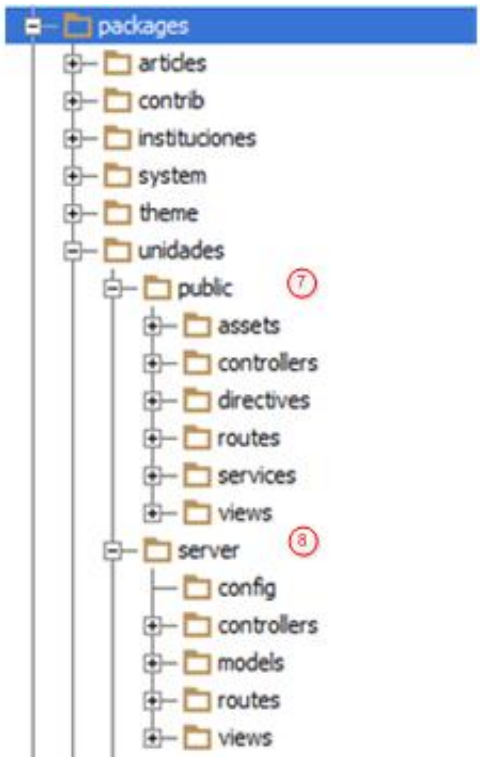


Fig. 5 Comando de creación de packages

Una vez creado el / los paquetes, serán posicionados en un nivel superior de este directorio, es decir si el package unidades se encuentra en la ruta: **/packages/custom/unidades**, la nueva ruta que deberá tener será: **/packages/unidades** y se deberá de eliminar el directorio **/custom**, esto se realizara independiente de la plataforma en la cual se trabaje.



La acción de reposicionar el package creado se debe a que de esta manera el package permanecerá visible para otros packages creados por el usuario o por los packages de contribución de la pila de aplicación de MEAN.IO, como el package wixmedia, taken y otros de la comunidad.

La **Fig. 6**, ilustra la taxonomía de un package del tipo **/custom,** el cual está dividido principalmente en 2 directorios **/public** y **/server**. El directorio /public alberga los componentes de angular que básicamente representa el backend de la aplicación.

El directorio **/public/ assets (7)** contiene las .js y .ccs del package local en este caso **/unidades**.

/public/controllers, definen el contenido de los controladores de angular los cuales mediante la variable **$scope**, determinan el estado real de los componentes para el databinding de la aplicación.

Fig.6 Directorios /public y /server

/public/directives, son los encargados de brindar funcionalidades extras a los componentes de los formularios como la conversión de datos de string a number. **/public/routes**, son archivos encargados de identificar las rutas y los archivos html encargados de responder a las peticiones emitidas en las rutas.

/public/services, contienen los archivos .js encargados de enlazar la capa del modelo con los controladores, alojando los servicios que con inyección de dependencia se hacen visibles en los controladores.

/public/views, continen los htmls que tienen la funcionalidad de la aplicación, los cuales están provistos de bootstrap.

El directorio **/server (8)** contienen los archivos y directorios vinculados con el backend de la aplicación, **/server/controllers,** alojan los controladores de la aplicación, estos controladores son los encargados de brindar funcionalidad al modelo de la aplicación (base de datos).

/server/routes, alojan los archivos encargados de gestionar el servicio CRUD, para la aplicación, la petición de servicios realiza solicitudes a la capa del modelo y los routes responden con las rutas de los servicios del lado del modelo.

/server/models, alojan los archivos .js de las colecciones y documentos de la base de datos mongodb, los cuales cuentan con restricciones de validación de campos y atomicidad de datos en la base de datos.

Una vez explicado el contenido de los directorios de la aplicación, se procederá a pasar a producción la aplicación.

Producción de la aplicación Transur

El primer paso es poseer una tarjeta de compra o tarjeta de crédito de cobertura internacional, para efectos de realizar una transferencia electrónica con la cuenta del hosting a contratar. Las opciones de hosting no son extensas para la pila de aplicaciones MEAN.IO, por lo que se decidió por el servicio de hosting de google.

Una vez ingresada y validada una cuenta de facturación en Google Cloud Platform, se procede a crear una instancia del servidor Debían preconfigurado, como muestra la Fig. 7

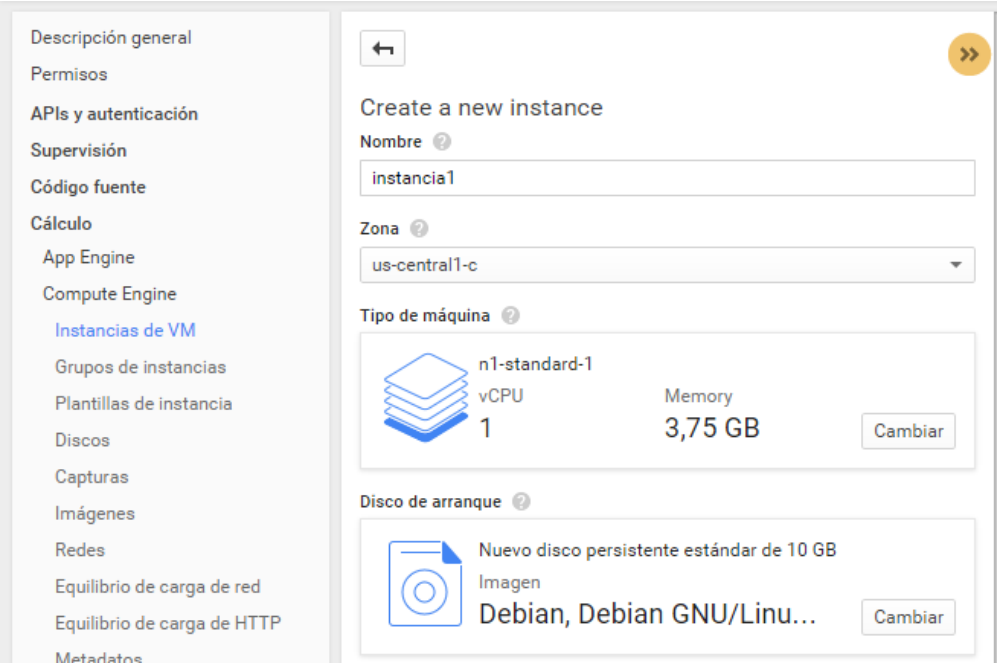


Fig. 7: Instancia del Servidor DEBIAN GNU/Linux

Se inicia una nueva instancia y se elige la zona horaria del servidor, DEBIAN GNU/Linux

Fig. 8 Monitoreo de Instancia del Servidor

En la figura anterior se muestra la instancia creada con los siguientes parámetros: Nombre, Zona, Disco, Red, Usadas por: IP Externa.

El icono IP Externa es seleccionado, y muestra un menú de tareas, donde se crea una nueva tarea con el nombre: mean y puerto output: tcp: 3000, se guarda la configuración y la página de bienvenida del hosting se refleja en el navegador.

Fig. 9 Deploy de la aplicación MEAN.IO

Una vez probado el flamante hosting, se debe de instalar una maquina Linux local, en cualquiera de sus versiones o en todo caso una máquina virtual.

Para la realización del presente manual se utilizó una maquina: **ubuntu-gnome-15.04-desktop-i386** con una CPU de doble núcleo de 2.90 Ghz y 1 Gb de memoria RAM.

Para iniciar abrimos un terminal con la cuenta local, para abrir una consola de administrador se procede a tipear el comando: **sudo bash** tal como lo ilustra la Fig. 10

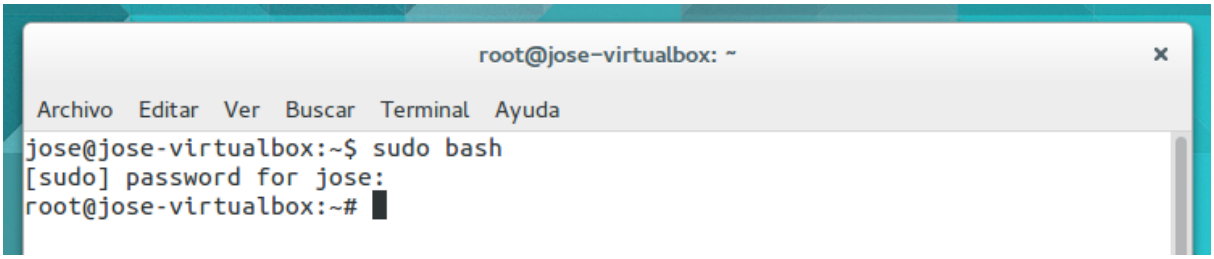


Fig. 10 Habilitando permisos de consola de administrador

Una vez iniciado el terminal con el usuario root se crea una public key para acceder al servidor de Google Cloud Platform, haciendo uso del comando: **ssh-keygen –b 4096 –t rsa**

Fig. 11 Generando una clave SSH

Una vez generada la clave se la puede mostrar en consola con el comando: **$ cat ~/.ssh/id_rsa.pub**

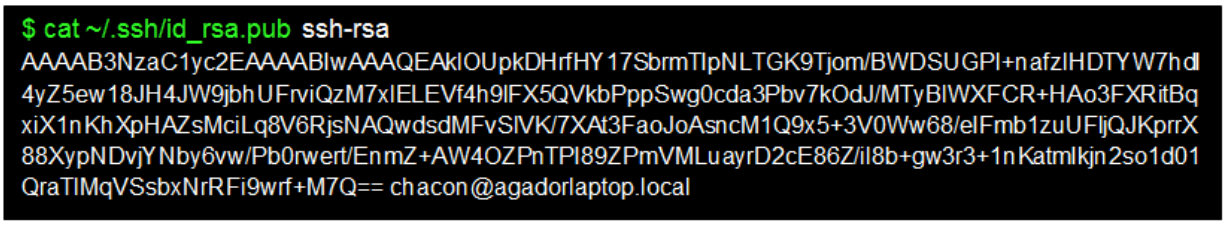


Fig. 12 Clave pública ssh

Se deberá de copiar el contenido de la clave ssh publica de la maquina local y se lo deberá de agregar en la lista de claves ssh del servidor, donde reside la aplicación, la lista de claves de la aplicación se encuentra en los detalles de información de la instancia del servidor como lo muestra la Fig. 13.

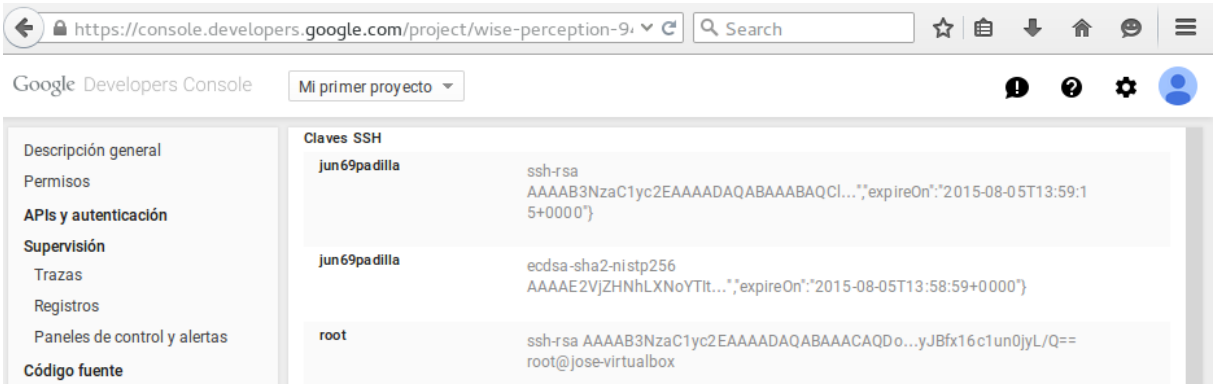


Fig. 13 Claves ssh en el servidor

La generación y adhesión de la clave ssh de la maquina local nos permitirá comunicarnos sin logueo al servidor de la aplicación.

La Fig. 14, muestra la comunicación de nuestra maquina local con el servidor remoto de la aplicación Transur con el comando del protocolo **ssh**.

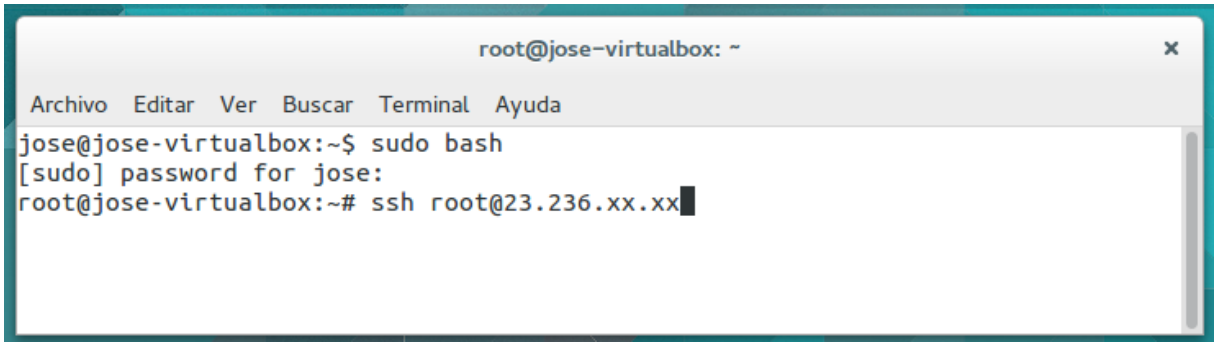


Fig.14 Comunicación con el servidor remoto

Una vez conectado con el servidor se tendrá un terminal con el **prompt** de la Fig. 15, el cual indica el usuario y la instancia del servidor al que está conectado **root@mean-6m2r**

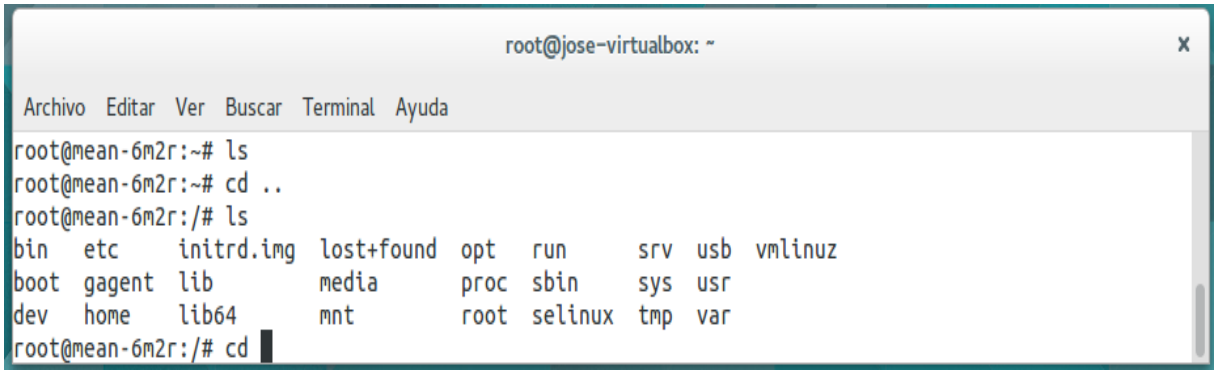


Fig. 15 Consola del servidor remoto

Una vez habilitada la consola del administrador se procederá a abrir otra terminar para copiar los archivos locales al servidor, en el **path** correspondiente en relación a los directorios del servidor remoto. Se lo hace con la ayuda del comando **scp** como muestra la Fig. 16.

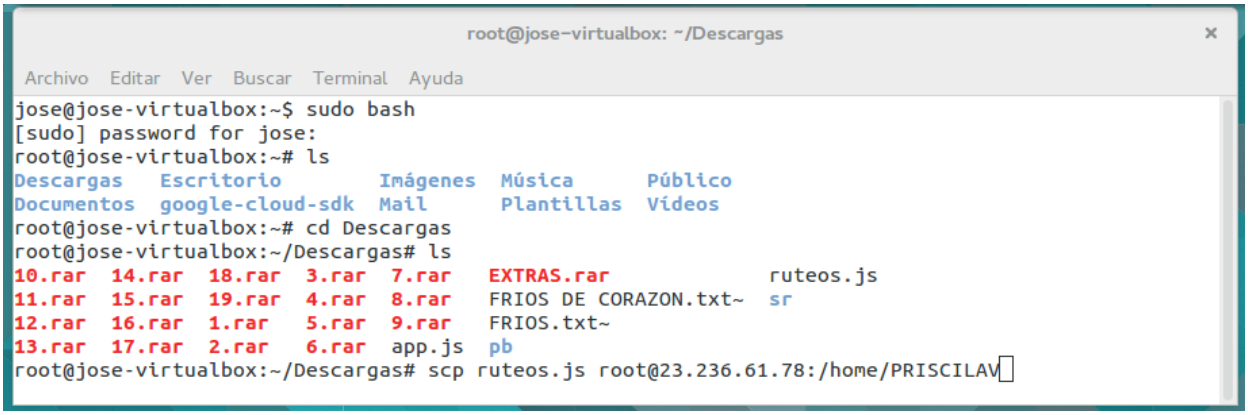


Fig. 16 Copia del archivo en el servidor remoto

En el caso de existir archivos extras como por ejemplo la generación de reportes como es el caso de FileSaver.js deberá de ser ubicado en el directorio correspondiente con la respectiva actualización de listado del archivo de configuración **assets.json**.

El contenido del manual técnico es muy denso, pero además es muy conciso de manera que el / los programadores de MEAN.IO, puedan realizar las actualizaciones necesarias y publicarlas en muy poco tiempo.

Como se mencionó en las conclusiones los encargados de las actualizaciones del sistema Transur deberán de tener conocimientos intermedios de JavaScript, Node.js, Express, Angular, Mongoose, Mongodb, Html 5, Css, Bootstrap y administración de Linux

Los archivos de configuración y de la aplicación Transur se encuentran en el cd adjunto, al igual que el manual técnico y el manual de usuario.