


## **Guía de Cloud V2 (Referencia proyecto base)**

## Historial de Revisiones

Fecha	Versión	Descripción	Autor
19/01/2023	1.0	Creación del documento	Lucio Chumil
26/01/2023	1.1	Actualización del documento	Lucio Chumil

## Contenido

<b>HISTORIAL DE REVISIONES.....</b>	<b>2</b>
<b>CONTENIDO.....</b>	<b>3</b>
<b>1. DESCRIPCIÓN GENERAL.....</b>	<b>4</b>
<b>2. DIRECCIONES .....</b>	<b>4</b>
<b>3. CONFIGURACIONES.....</b>	<b>5</b>

	Guía Cloud V2	<b>CSII – SIAMP</b> <b>Fecha:</b> 20/01/2023
	SIAMP	

## 1. Descripción General

Guía general para la implementación de nuevo servicio para la subida y descarga de archivos correspondiente al Cloud Versión No2, adicionalmente se detalla la reestructuración de servicios, controladores y eliminación de variables que ya no son útiles para dichos sistemas.


## 2. Direcciones

URL DEV: <https://cloud-de.mp.gob.gt>

URL STAGING: <https://cloud-st.mp.gob.gt>

URL PROD: <https://cloud.mp.gob.gt>

Documentacion: <https://cloud-de.mp.gob.gt/api-docs/>

	Guía Cloud V2	<b>CSII – SIAMP</b> <b>Fecha:</b> 20/01/2023
	SIAMP	

### 3. Configuraciones

#### Backend

**Variables de entorno:** Eliminación de variables marcadas en rojo y agregar nueva ruta en archivos env marcadas en verde.

##### .env development


```
#CLOUD
- CLOUD_HOST      =http://meandesa.mp:41020
- CLOUD_SYSTEM    =42024
-
-
- #Servicios de Oauth 1.0
- OAUTH_URL_TOKEN= http://meandesa.mp:42024/oauth/token
- OAUTH_TOKEN_USER= downloadcloud
- OAUTH_TOKEN_PASSSER= J-*sq[S55h,L;,S5LS_p?, :g4b'FC}gx8xpQ$-ky;]a+3am'([NFtY:'w5,T[5
-
+ CLOUD_HOST      =https://cloud-de.mp.gob.gt
```

```
#CLOUD
CLOUD_HOST      =https://cloud-de.mp.gob.gt
```

##### .env staging

```
#CLOUD
- CLOUD_HOST      =http://meandesa.mp:41020
- CLOUD_SYSTEM    =42024
-
-
- #Servicios de Oauth 1.0
- OAUTH_URL_TOKEN= http://meandesa.mp:42024/oauth/token
- OAUTH_TOKEN_USER= downloadcloud
- OAUTH_TOKEN_PASSSER= J-*sq[S55h,L;,S5LS_p?, :g4b'FC}gx8xpQ$-ky;]a+3am'([NFtY:'w5,T[5
-
+ CLOUD_HOST      https://cloud-st.mp.gob.gt
```

```
#CLOUD
CLOUD_HOST      =https://cloud-st.mp.gob.gt
```

	Guía Cloud V2	CSII – SIAMP Fecha: 20/01/2023
	SIAMP	

## .env production

```
#CLOUD
- CLOUD_HOST      =http://meandesa.mp:41020
- CLOUD_SYSTEM    =42024
-
-
- #Servicios de Oauth 1.0
- OAUTH_URL_TOKEN= http://meandesa.mp:42024/oauth/token
- OAUTH_TOKEN_USER= downloadcloud
- OAUTH_TOKEN_PASSSER= J-*sq[S55h,L;,S5LS_p?, :g4b'FC}gx8xpQ$-ky;]a+3am' ( [NftY: 'w5,T[5
-
+ CLOUD_HOST      = https://cloud.mp.gob.gt
```

```
#CLOUD
CLOUD_HOST      =https://cloud.mp.gob.gt
```

Limpieza de variables en [archivo cf.js](#)

```
- //CLOUD
- cloudHost: process.env.CLOUD_HOST,
- cloudSystem: process.env.CLOUD_SYSTEM,
- cloudHostNuevo: process.env.CLOUD_HOST_NUEVO,
+ //CLOUD
+ cloudHost: process.env.CLOUD_HOST,

// OAUTH
- oauthUrlToken: process.env.OAUTH_URL_TOKEN,
- oauthTokenUser: process.env.OAUTH_TOKEN_USER,
- oauthTokenPass: process.env.OAUTH_TOKEN_PASSSER,
```

Modificación de servicio *download* en *cloud.services.js*

```
exports.download = async (idCloud, obj) => {

  const url = cfg.cloudHost + "/file/" + idCloud;

  const configAxios = {
    headers: {
      'cui': obj.cui,
      'system': obj.system,
      'Authorization': obj.token,
    },
    responseType: "stream",
  };

  try {

    const response = await axios.get(url, configAxios);

    let date = new Date().getTime();
    const path = Path.resolve(__dirname, "../..", "tmp", idCloud + date);

    let writer = null;
    try {
      writer = Fs.createWriteStream(path);
    } catch (err) {
      console.log(err);
    }

    response.data.pipe(writer);
    new Promise((resolve, reject) => {
      writer.on("finish", resolve);
      writer.on("error", reject);
    });

    return responsesServices.success({
      type: response.headers["content-type"],
```

```
size: response.headers["content-length"],
name: response.headers["x-filename"],
path: writer.path,
});
} catch (error) {
  console.log(error);
  return responsesServices.error(error);
}
};
```

Modificación de controlador *download* en *cloud.client.js*

```
exports.download = async (req, res) => {
  // realizar logica del negocio
  const idCloud = req.body.idCloud;
  const headers = {
    token: req.headers.authorization,
    cui: req.headers.cui,
    system: req.headers.system
  }

  const responseFiledownloaded = await download(idCloud, headers);

  if (responseFiledownloaded.valid) {
    // Obtener archivo en base 64
    await delay(100);
    const base64 = await new Promise((resolve, reject) => {
      Fs.readFile(responseFiledownloaded.data.path, { encoding: "base64" }, (err, data) => {
        if (err) {
          reject(err);
        } else {
          // Construyendo respuesta
          const file = {
            type: responseFiledownloaded.data.type,
            size: responseFiledownloaded.data.size,
            name: responseFiledownloaded.data.name,
            base64: data,
          };
        }
      });
    });
  }
};
```



```
resolve(res.status(200).send(responsesServices.success((data = file), (msg =
successMessages.SUCCESS_FIND))));
    }
  });
});


// Eliminar archivo temporal
try {
  Fs.unlinkSync(responseFiledownloaded.data.path);
} catch (e) {
  console.log("PROBLEMA AL ELIMINAR ARCHIVO TEMPORAL");
  console.log(e);
}
} else {
  // retornar mensaje de error
  res.status(200).send(responsesServices.error((msg = errorMessages.ERROR_DOWNLOAD));
}
};
```

Eliminación de los siguientes controladores en [cloud.client.js](#)

```
> exports.downloadPdf2 = async (req, res) => { ...
};

> exports.downloadPdf = async (req, res) => { ...
};
```

Eliminación de rutas innecesarias.

	Guía Cloud V2	CSII – SIAMP Fecha: 20/01/2023
	SIAMP	

```

- /**
-  * @swagger
-  * /archivo/downloadPdf:
-  *   post:
-  *     summary: "Descargar archivo PDF"
-  *     description: Descarga un archivo de Cloud por medio de su id
-  *     tags: [ARCHIVO]
-  *     responses:
-  *       200:
-  *         description: Ok
-  *       203:
-  *         description: No autorizado.
-  *       400:
-  *         description: Error en la validación o Error en los Headers.
-  */
- app.get("/archivo/downloadPdf/:idCloud", CLOUD.downloadPdf);
-

```

Agregamos seguridad a la siguiente ruta:

```

/**
* @swagger
* /archivo/download:
*   post:
*     summary: "Descargar archivo"
*     description: Descarga un archivo de Cloud por medio de su id
*     tags: [ARCHIVO]
*     security:
*       - Bearer: []
*     parameters:

```

```
* - $ref: "#/parameters/systemHeader"
* - $ref: "#/parameters/cuiHeader"
* - $ref: "#/parameters/rolHeader"
* responses:
*   200:
*     description: Ok
*   203:
*     description: No autorizado.
*   400:
*     description: Error en la validación o Error en los Headers.
*/
app.post("/archivo/download", oauthController.verify, CLOUD.download);
```

## Frontend:

**Variables de entorno:** Eliminación de variables marcadas en rojo y agregar nueva ruta en archivos env marcadas en verde.

**.env development**

```
#INICIAR CON yarn install, yarn start, SI DA ERRORES EJECUTAR NPM LINK Y LUEGO yarn start
- REACT_APP_URL_DOWNLOAD_FILE = http://172.18.141.171:10532/archivo/downloadPdf/

- #REACT_APP_URL_DOWNLOAD_FILE = http://localhost:#10123/archivo/downloadPdf/
-
-
- #REACT_APP_CLOUD_URL = http://meandesa.mp:41020/cloud/file/save/one
+ REACT_APP_CLOUD_URL = https://cloud-de.mp.gob.gt
- REACT_APP_CLOUD_URL_DOWNLOAD = http://meandesa.mp:41020/cloud/file/
```

```
REACT_APP_CLOUD_URL = https://cloud-de.mp.gob.gt
REACT_APP_CLOUD_IDSISTEM = sftp_pruebas
```

## .env staging

```
#Servicio CLOUD
REACT_APP_CLOUD_IDSISTEM = sftp_pruebas
- REACT_APP_CLOUD_URL = http://meandesa.mp:41020/cloud/file/save/one
- REACT_APP_CLOUD_URL_DOWNLOAD = http://meandesa.mp:41020/cloud/file/
- REACT_APP_URL_DOWNLOAD_FILE = http://172.18.141.171:10356/archivo/downloadPdf/
+ REACT_APP_CLOUD_URL = https://cloud-st.mp.gob.gt
```

```
REACT_APP_CLOUD_URL = https://cloud-st.mp.gob.gt
REACT_APP_CLOUD_IDSISTEM = sftp_pruebas
```

	Guía Cloud V2	<b>CSII – SIAMP</b> <b>Fecha:</b> 20/01/2023
	SIAMP	

### .env production

```
#servicio CLUUD
REACT_APP_CLOUD_IDSYSTEM = cupones
- REACT_APP_CLOUD_URL = https://api.mp.gob.gt/cloud/cloud/file/save/one
- REACT_APP_CLOUD_URL_DOWNLOAD = https://api.mp.gob.gt/cloud/cloud/file/
- REACT_APP_URL_DOWNLOAD_FILE = https://api.mp.gob.gt/cupones/archivo/downloadPdf/
+ REACT_APP_CLOUD_URL = https://cloud.mp.gob.gt
```

```
REACT_APP_CLOUD_URL = https://cloud.mp.gob.gt
REACT_APP_CLOUD_IDSYSTEM = cupones ←-identificador de cada sistema
```

Limpieza de variables en el archivo [index.js](#)

```
cloudIdSystem: process.env.REACT_APP_CLOUD_IDSYSTEM,
cloudUrl: process.env.REACT_APP_CLOUD_URL,
- cloudUrlDownload : process.env.REACT_APP_CLOUD_URL_DOWNLOAD,
```

Modificación de función de [handleUploadFile](#) en [UploadFile.js](#)

```
const handleUploadFile = async () => {  
  var files = uploadFile;  
  var referenciald = 0  
  
  if (files.length > 0 && props.TABLA_REFERENCIA !== undefined) {  
  
    if (props.TABLA_REFERENCIA_ID !== undefined) {  
      referenciald = props.TABLA_REFERENCIA_ID;  
    }  
    setUploadPercentage(true)  
    const niplogin = (info.token == null) ? "" : info.token.user.info.nip;  
  
    let idSystem = urls.cloudIdSystem  
    let directory = niplogin  
  
    var ArchivosSelected="";  
  
    for (var i = 0; i < files.length; i++) {  
  
      ArchivosSelected=ArchivosSelected+' , '+files[i].name  
  
      const form = new FormData()  
      form.append('idSystem', idSystem);  
      form.append('directory', directory);  
      form.append('file_to_save', files[i]);  
      var config = {  
        maxContentLength: 1024 * 1024 * 1024 * 10  
      }  
  
      try {  
        const url = urls.cloudUrl + '/file/save/one';  
        const response = await axios.post(url, form, config)  
        if (response.data.valid) {  
          let data = response.data.data;  
          let ObjAdd = {  
            "idCloud": data.idFile,  
            "nombre": files[i].name,  
            "tipo": files[i].type,  
            "tamaño": files[i].size,  
            "tablaReferencia": props.TABLA_REFERENCIA,  
            "tablaReferenciald": referenciald,  
            "infoUser": info.token.user.info  
          }  
  
          let objRespuesta = await service.addArchivo(ObjAdd);  
        }  
      }  
    }  
  }  
}
```

```
        if (objRespuesta.valid) {
            showAlertMensaje({
                open: true,
                message: "Subió correctamente el archivo.",
                alertType: "success"
            })
        } else {
            showAlertMensaje({
                open: true,
                message: "Error consulte con el administrador.",
                alertType: "error"
            })
        }
        setUploadFile(null)
        setVerCarga(false)
        //setFileInfo({})
        setFilesSelected("")
        setUploadPercentage(false)
    }
    else {
        showAlertMensaje({
            open: true,
            message: "Error consulte con el administrador.",
            alertType: "error"
        })
    }
}
catch (error) {
    showAlertMensaje({
        open: true,
        message: "Error consulte con el administrador.",
        alertType: "error"

    })
}
}
setFilesSelected(ArchivosSelected)
//////////////////// FIN DE FOR
} else {
    showAlertMensaje({
        open: true,
        message: "Error consulte con el administrador.",
        alertType: "error"
    })
}
```

```
}  
}
```

Modificar la función `handleHref` en descarga de archivo en `UploadFileList.js`

```
const handleHref = async (idCloud) => {  
  const aux = {idCloud}  
  let objRespuesta = await Service.download(aux);  
  
  if (objRespuesta.valid) {  
    Service.descargaDirecta(objRespuesta.data);  
  } else {  
    showAlertMensaje({  
      open: true,  
      message: objRespuesta.msg ? objRespuesta.msg : "Error, consulte con el  
      administrador. Descarga.",  
      alertType: "error",  
    });  
  }  
  
}
```

Y al momento de hacer el llamado solo enviar el `idCloud` en el mismo archivo de `UploadFileList.js`

```
handleHref(r.idCloud)
```

Agregar función en componentes globales en `Service.js`

```
export function descargaDirecta(obj) {  
  const linkSource = `data:${obj.type};base64,${obj.base64}`;  
  const downloadLink = document.createElement("a");  
  const fileName = obj.name;
```



```
downloadLink.href = linkSource;  
downloadLink.download = fileName;  
downloadLink.click();  
}
```