

# INFORME



**APLICACIÓN WEB PARA LA ADMINISTRACION DE PERSONAL  
DEL GOBIERNO AUTONOMO DEPARTAMENTAL DE CHUQUISACA**

**06 de Julio de 2018**

Bolivia - Chuquisaca – Sucre



**AUTOR:** ALAN WALTER MACHUCA DUREX <[alanwalter45@gmail.com](mailto:alanwalter45@gmail.com)>  
**TECNICO I DE LA DIRECCIÓN GENERAL DE RECURSOS HUMANOS**

## CONTENIDO

RESUMEN.....	1
INTRODUCCIÓN.....	1
SITUACIÓN ACTUAL .....	1
OBJETIVO GENERAL.....	2
OBJETIVOS ESPECIFICOS .....	2
ANÁLISIS FODA - SWOT (STRENGTHS WEAKNESSES OPPORTUNITIES) .....	2
APLICACIÓN DE ADMINISTRACIÓN DE PERSONAL.....	3
ESTRUCTURA OPERACIONAL DE RECURSOS HUMANOS .....	3
REQUISITOS DE CONTRATACIÓN .....	4
.....	5
ALCANCE FUNCIONAL .....	6
PERFILES DE USUARIO .....	6
DESCRIPCIÓN DE LOS PERFILES .....	7
RESOLUCIÓN AL PROBLEMA .....	10
HERRAMIENTAS .....	10
.....	12
REQUISITOS DE SOFTWARE Y HARDWARE .....	13
DURACIÓN - TIEMPOS .....	15
EQUIPO DE TRABAJO REQUERIDO.....	15
AVANCE EN EL PROYECTO .....	15
PROTOTIPADO DE SOFTWARE.....	19
RECURSOS FACILITADOS – INFORMACIÓN .....	19
TECNOLOGÍAS UTILIZADAS .....	20
CARACTERÍSTICAS DEL SISTEMA .....	25
METODOLOGÍA DE DESARROLLO .....	31
CONCLUSIONES.....	31



## RESUMEN

La aplicación web para la administración de personal está previsto ejecutarse en la Dirección General de Recursos Humanos del Gobierno Autónomo Departamental de Chuquisaca y estará básicamente diseñado para gestionar la información general y extra de los Servidores Públicos que forman parte integrante de nuestra entidad.

## INTRODUCCIÓN

El Gobierno Autónomo Departamental de Chuquisaca recibe diferentes tipos de servidores públicos trabajando áreas distintas y en común, es bastante razonable llevar un control de esta información, por lo que se propone, sentar una lógica base, para poder atender esta problemática. Considerando tecnologías de la información para satisfacer este requerimiento.

Motivo por el cual, este documento pretende mostrar una posible solución.

## SITUACIÓN ACTUAL

- La Dirección General de Recursos Humanos maneja un gran volumen de información relacionada a los servidores públicos, contando actualmente con un aproximado de 410 personas con ITEM y 297 personas con CONTRATO.
- Los diferentes trámites administrativos en la gestión de personal, se tornan muchas veces pesados, rutinarios y burocráticos.
- La información que generan los servidores públicos de la Dirección General de Recursos Humanos funciona de manera separada, obligando a realizar el cruce permanente de información, situación que dificulta la generación de información permanente, oportuna y confiable.

## OBJETIVO GENERAL

Desarrollar una solución (**aplicación web**) que permita gestionar la información registrada por la ADMINISTRACIÓN DE PERSONAL en la Dirección General de Recursos Humanos.

## OBJETIVOS ESPECIFICOS

- Emisión de reportes
- Registro de información general
- Registro de información extra ( política, afiliación, etc)
- Búsquedas de archivos adjuntos
- Gestiones de trabajo
- Seguridad – regular
- Configuración en despliegue
- Crear API – Administración de personal

Antes de empezar a describir el aplicativo como tal, empezaremos por mencionar y aclarar que las tecnologías sugeridas para la construcción de esta aplicación son para establecer un software fiable y reutilizable para futuras infraestructuras u orquestación de procesos o cruce de información, además tomando que como principal objetivo del aplicativo será, formar parte integrante de un sistema de recursos humanos.

Motivo por el cual será muy importante construir un entorno de software adecuado.

## ANALISIS FODA - SWOT (STRENGTHS WEAKNESSES OPPORTUNITIES)

Análisis rápido sobre la administración de personal considerando los indicadores de toma de decisión FORTALEZAS DEBILIDADES, AMENAZAS y OPORTUNIDADES

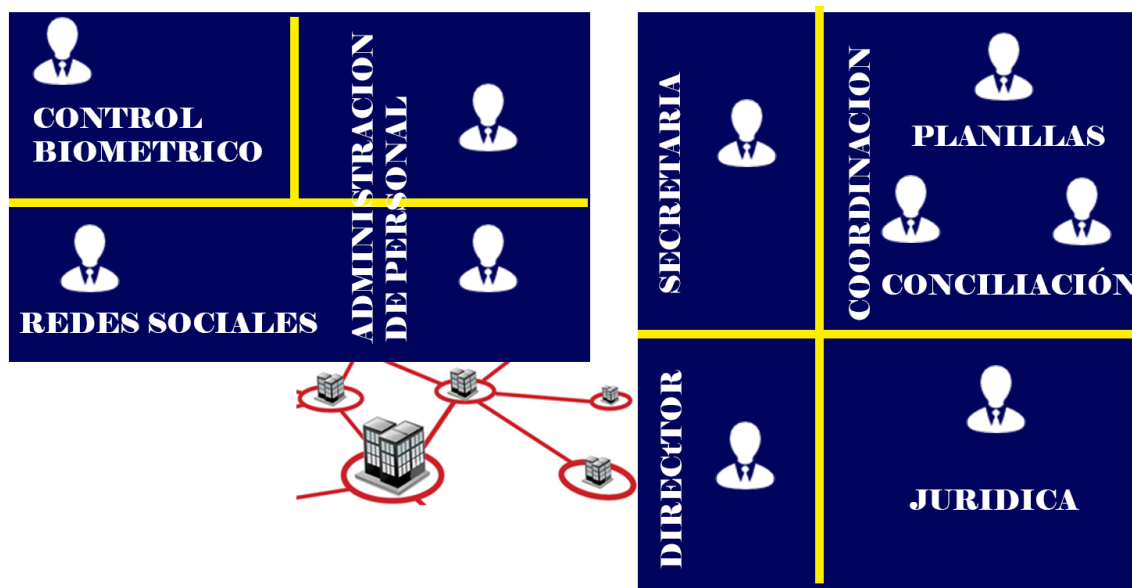
ORIGEN	FORTALEZAS	DEBILIDADES
INTERNO	<ul style="list-style-type: none"> <li>• Seguridad: Tokens, bcrypt, hash,md5</li> <li>• Sincronización automática</li> <li>• Rapidez – velocidad</li> <li>• Portabilidad</li> </ul>	<ul style="list-style-type: none"> <li>• Duplicación de información</li> <li>• Tiempo de espera en búsquedas</li> </ul>
	OPORTUNIDADES	AMENAZAS
EXTERNO	<ul style="list-style-type: none"> <li>• Compartir información</li> <li>• Software Portable</li> <li>• Reducir costos</li> <li>• Reutilizable</li> <li>• Reducir tiempos</li> </ul>	<ul style="list-style-type: none"> <li>• Ataques informáticos</li> <li>• Perdida de datos</li> <li>• Falla de dispositivos, red u otros</li> </ul>

## APLICACIÓN DE ADMINISTRACIÓN DE PERSONAL

El servidor público es el foco principal para este aplicativo, por lo que resulta de vital importancia para el G.A.D.C.H el contar con una BASE DE DATOS de todo el personal, para que pueda ser utilizada por las diferentes fuentes de trabajo de la Dirección General de Recursos Humanos, esta pieza de software se constituye como información base para el sistema de “**recursos humanos**”.

## ESTRUCTURA OPERACIONAL DE RECURSOS HUMANOS

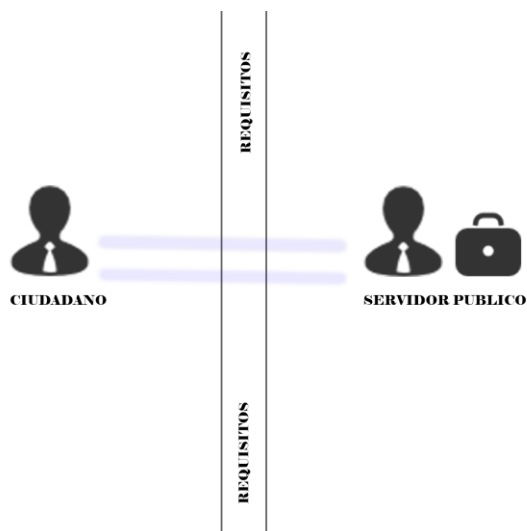
La siguiente imagen describe las divisiones laborales del personal de recursos humanos



## RECURSOS HUMANOS

### REQUISITOS DE CONTRATACIÓN

Todo ciudadano que quiera ocupar un cargo dentro del gobierno autónomo departamental de Chuquisaca deberá cumplir con requisitos de contratación, según su cargo y jerarquía dentro de la estructura organizacional.





**SERVIDOR PUBLICO**

\* SECRETARIOS,  
DIRECTORES Y JEFES

\* TECNICOS I Y PROFESIONALES

\* TENICOS II Y III

\* ASISTENTES I Y II

\* AUXILIARES I Y II

\* CHOFERES

REQUISITOS



**PLANTILLA  
DE  
REQUISITOS**

REQUISITOS

Información general y extra generada por el servidor público



**SERVIDOR PÚBLICO**



**INFORMACIÓN + CURRICULUM**



**MEMORADUMS Y OTROS**



## **ALCANCE FUNCIONAL**

- Información general de los servidores públicos (nombre, edad, estado civil, género, domicilio, C.I., etc.)
- información de inamovilidad (Discapacidad, prenatal, lactancia, afiliaciones, etc.)
- información de afiliaciones (caja de salud, AFP)
- Información de la contratación (cargo, ítem, contrato, nivel, etc.)
- Reportes generales
- Documentos presentados según normativa (Título, hoja de vida, SIPASE, Libreta Servicio Militar, etc.
- Vacaciones
- Felicitaciones
- Reportes

## **EXTENSIONES DE LA APLICACIÓN**

- Cumpleaños
- Creación de encuestas y actividades
- Publicaciones
- File Personal

## **PERFILES DE USUARIO**

Se ha visto conveniente habilitar los siguientes perfiles en el sistema, para lograr gestionar este de la mejor manera.

- 1. Servidor Público – Estándar**
- 2. Director General de Recursos Humanos**
- 3. Gobernador**
- 4. Jefes e inmediatos superiores**
- 5. Planillas/ITEM,CONTRATO**
- 6. Parte Legal / CONTRATACIONES**

- 7. **Secretaría**
- 8. **Control Biométrico**
- 9. **Administración de Personal**
- 10. **Sistemas**

## **DESCRIPCIÓN DE LOS PERFILES**

A continuación, se describe de manera resumida los distintos perfiles nombrados anteriormente

### **1.- Servidor Público – Estándar**

- Registro de su información general y específica solicitada
- Solicitar permisos de salida a través de la web
- Verificar su información
- Programación y reporte de actividades mensuales
- Consultas con la secretaría de recursos humanos
- Reportes

### **2.- Director General de Recursos Humanos**

- Cuenta con los permisos de un servidor público estándar (Perfil 1)
- Acceder a reportes generales de la información correspondiente a todos los servidores públicos de la institución
- Solicitudes de información al administrador de personal
- Cuenta con los permisos de un servidor público estándar (Perfil 4)

### **3.- Gobernador**

- Acceder a reportes generales de la información correspondiente a todos los servidores públicos de la institución
- Cuenta con los permisos de un servidor público estándar (Perfil 4)

#### **4.- Jefes e inmediatos superiores**

- Cuenta con los permisos de un servidor público estándar (Perfil 1)
- Autoriza o rechaza las solicitudes de salida del personal dependiente
- Aprueba o rechaza la programación de actividades mensuales del personal de su dependencia y supervisa su ejecución
- Acceder a reportes generales de la información correspondiente a todos los servidores públicos de su área dependiente

#### **5.- Planilla/ITEM-CONTRATO**

- Cuenta con los permisos de un servidor público estándar (Perfil 1)
- Visualiza y supervisa la información de datos de los servidores públicos de la gobernación: asistencia, bajas médicas, retiros justificados, renunciaciones, etc.
- Notifica sobre observaciones detectadas
- Reportes

#### **6.- Parte Legal / CONTRATACIONES**

- Cuenta con los permisos de un servidor público estándar (Perfil 1)
- Visualiza y supervisa la información de datos de los servidores públicos de la gobernación: ingresos, bajas.
- Notifica sobre observaciones detectadas
- Reportes

#### **7.- Secretaría (Asistente)**

- Cuenta con los permisos de un servidor público estándar (Perfil 1)
- Realiza publicaciones en la página web.
- Actualiza datos de su unidad: Teléfonos, ubicaciones

## **8. Control Biométrico**

- Cuenta con los permisos de un servidor público estándar (Perfil 1)
- Control de Asistencia
- Supervisión de los formularios de salida
- Supervisión de la Programación Mensual Individual de los servidores públicos
- Registro de regularizaciones de asistencia por diferentes motivos

## **9.- Administración de Personal**

- Cuenta con los permisos de un servidor público estándar (Perfil 1)
- Registra toda la información clasificada referente a los recursos humanos de la institución descrita en el Módulo 1 y genera la BASE DE DATOS

## **10.- Sistemas**

- Cuenta con los permisos de un servidor público estándar (Perfil 1)
- Realiza los ajustes necesarios en el sistema para que este funcione de la mejor manera
- Asignación de perfiles
- Crea respaldos backups

## RESOLUCIÓN AL PROBLEMA

MAQUETACIÓN	LENGUAJE DE PROGRAMACIÓN	ESTILOS
HTML5	JAVASCRIPT (ES6) SQL	CSS3

FRAMEWORKS	PLATAFORMA	MOTOR DE BASE DE DATOS
ANGULAR EXPRESS JS	NODEJS y NPM	MYSQL SQLITE
		FIREBIRD

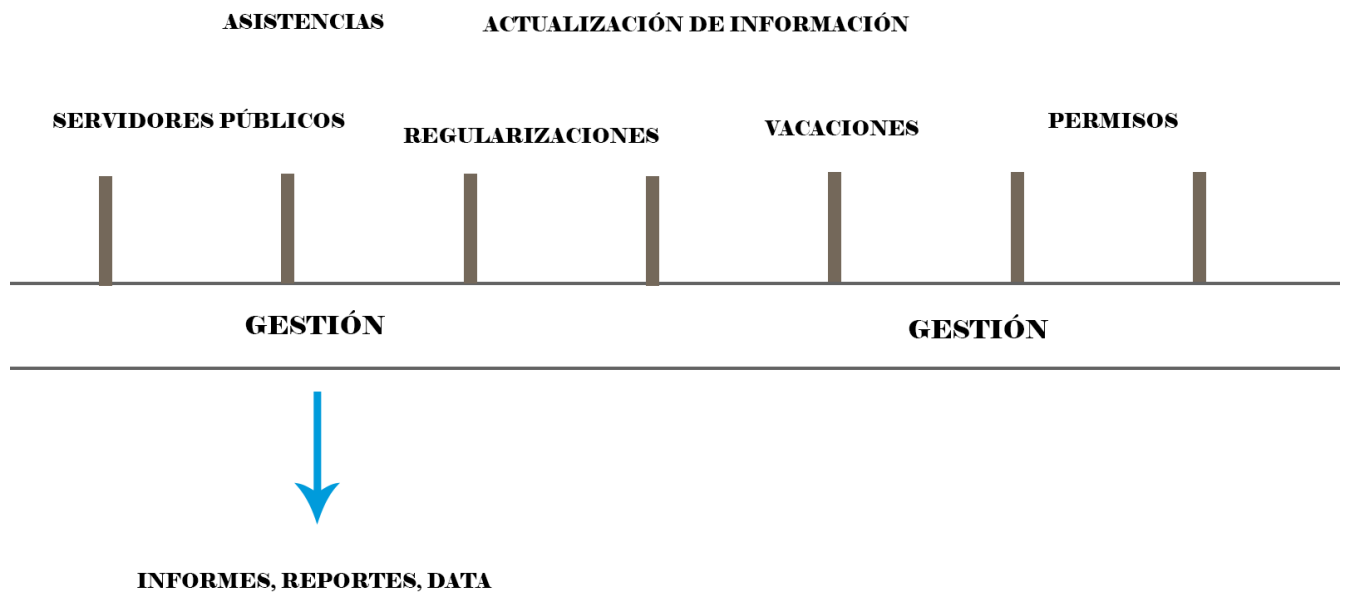
## HERRAMIENTAS

- 1.- VISUAL STUDIO CODE
- 2.- POSTMAN
- 3.- GOOGLE CHROME / MOZILLA FIREFOX / OPERA
- 4.- WORKBENCH
- 5.- FLAMERIBON
- 6.- SQLITESTUDIO
- 7.- ANDROID STUDIO
- 8.- SH SERVER
- 9.- OPEN OFFICE, MICROSOFT OFFICE

## VOLUMEN DE INFORMACIÓN GENERADA

Los siguientes gráficos muestran los indicadores a tomar en cuenta para la predicción del flujo de datos y la proliferación de la información.

### EVENTOS – SUCESOS





SERVIDORES  
PÚBLICOS

2017

2018

2019

2020

LINEA DE TIEMPO

## MOVIMIENTOS MENSUALES

ENERO

FEBRERO

MARZO

ABRIL

LINEA DE TIEMPO

# SERVIDORES PÚBLICOS

## REQUISITOS DE SOFTWARE Y HARDWARE

Para establecer un entorno de desarrollo y despliegue de software, se necesita cumplir ciertos requisitos que se detallan a continuación.

### REQUISITOS DE HARDWARE USUARIO

#### SISTEMA OPERATIVO

- Windows 7 +
- Ubuntu 16 +
- MacOS Leopard +
- Memoria Ram: 512 mb +
- Espacio en disco: 1gigas +

#### NAVEGADOR WEB

- google Chrome 30 ó posterior
- Mozilla Firefox 30 ó posterior
- opera 45 ó posterior

### REQUISITOS PARA DESPLIEGUE DEL SISTEMA

#### MINIMO

- 1 servidor Web. // **Hospedador de aplicaciones.**
- 1 estabilizador de energía eléctrica.



## NORMAL

- Servicio de Internet // **Internet**
- 8 Routers. // **Cobertura inalámbrica.**
- Cableado de red. // **Cobertura por cable.**
- 8 Dispositivos Biométricos. // **Control de asistencia.**
- 1 Servidor Web. // **Hospedador de aplicaciones.**
- 1 Portátil. // **Terminal de Mantenimiento.**
- 1 HDD externo // **Copias de respaldo – Backups.**
- 8 Cámaras de seguridad. // **Seguridad.**
- Sistema de alimentación ininterrumpida de energía eléctrica. // **Alimentación eléctrica UPS para el servidor.**

## REQUISITOS PARA EL SERVIDOR DE APLICACIONES

Servidor : **DELL PowerEdge T110 II**  
S.O : **Ubuntu Server 18.04**  
Disco duro : **500 gb**  
Memoria RAM: **1gb**

## MAQUINA UTILIZADA PARA EL DESARROLLO

Sistema Operativo : **Ubuntu 17.10**  
Memoria : **4gb**  
Procesador : **Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz**

Alan Walter Machuca Durex - [alanwalter45@gmail.com](mailto:alanwalter45@gmail.com)

## DURACIÓN - TIEMPOS

El tiempo fue calculado utilizando el COCOMO II para predicción en base a indicadores

TIEMPO EN MESES	TIEMPO EN DIAS	MARGEN DE ERROR
2	60	5%

## PROGRESO

TOTAL DIAS	TIEMPO AVANZADO	TIEMPO RESTANTE
60	45	15

## EQUIPO DE TRABAJO REQUERIDO

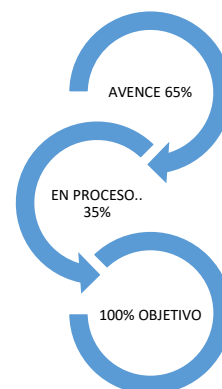
Personal necesario para construir la arquitectura de software ha soportar.

NORMAL		
1 DISEÑADOR DE APLICACIONES	1 ADMINISTRADOR DE BASE DE DATOS	1 DESARROLLADOR FULLSTACK
REGULAR		
1 ADMINISTRADOR DE BASE DE DATOS	1 DESARROLLADOR FULLSTACK	
MINIMO		
1 DESARROLLADOR FULLSTACK		

## AVANCE EN EL PROYECTO

Según evaluación de la capas, módulos y prototipos se puede estimar un alcance con 5% de margen de error.

65% AVANZADO  
35% EN PROCESO  
-----  
100% COMPLETADO



## SERVIDOR NODEJS

Captura del servidor corriendo en el puerto 3000

```
> node server/server.js

-----
      API - RECURSOS HUMANOS
-----
-| server: localhost:3000
-| platform:  linux x64
-| pid:   18124
-| startup: 2018-07-05 08:21:56
-----
      alanwalter45@gmail.com
-----
connect success...
█
```

## GENERACIÓN DEL APLICACIÓN CLIENTE EN ANGULAR

```
Date: 2018-07-06T00:22:22.535Z
Hash: ade55e7f53ce6f1a688c
Time: 28082ms
chunk {main} main.js, main.js.map (main) 670 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 227 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 5.4 kB [entry] [rendered]
chunk {scripts} scripts.js, scripts.js.map (scripts) 1.04 MB [rendered]
chunk {styles} styles.js, styles.js.map (styles) 240 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 5 MB [initial] [rendered]
i [wdm]: Compiled successfully.
█
```

## AUTENTICACIÓN

Menu Recursos Humanos - X +

localhost:4200/autenticacion

DIRECCIÓN GENERAL DE RECURSOS HUMANOS

**Gobierno Autónomo de CHUQUISACA**

\* USUARIO

mi usuario...

\* CONTRASEÑA

\*\*\*\*\*

Entrar

Bienvenido(a) al Sistema de Recursos Humanos, estamos felices de que seas parte de nuestro equipo de trabajo.

**QUE TENGAS UN EXCELENTE DÍA.**

¿Olvidaste tu contraseña ?

## RECUPERACIÓN DE CONTRASEÑA

Menu Recursos Humanos - X +

localhost:4200/autenticacion

DIRECCIÓN GENERAL DE RECURSOS HUMANOS

**RECUPERAR CONTRASEÑA**

\* CARNET DE IDENTIDAD

ej. 3423221

\* FECHA DE NACIMIENTO

mm/dd/yyyy

Recuperar Contraseña Cancelar

\* CONTRASEÑA

\*\*\*\*\*

Entrar

Bienvenido(a) al Sistema de Recursos Humanos, estamos felices de que seas parte de nuestro equipo de trabajo.

**QUE TENGAS UN EXCELENTE DÍA.**

¿Olvidaste tu contraseña ?

## VISTA PRINCIPAL

localhost:4200/perfil-publico/registros/informacion-general

DIRECCIÓN GENERAL DE RECURSOS HUMANOS

INICIO SALIR

actualizar reporte excel

**INFORMACIÓN GENERAL**

* CARNET DE IDENTIDAD 1037671	* CARNET EXPEDIDO CHUQUISACA	* NOMBRES FELIPE	* APELLIDO PATERNO ARANCIBIA
* APELLIDO MATERNO TANGO	* ESTADO CIVIL	* GENERO MASCULINO	* DIRECCIÓN CALLE LLAWARWAKAJ N°150 I
* TELEFONO FIJO telefono aquí...	* TELEFONO MOVIL 71169433	* FECHA DE NACIMIENTO 04/11/1947	* PAIS DE NACIMIENTO BOLIVIA
* DEPARTAMENTO CHUQUISACA	* PROVINCIA	* MUNICIPIO DE NACIMIENTO	* CORREO ELECTRONICO ejemplo@gmail.com

copyright © 2018 - alanwalter45

## REPORTE

Menu Recursos Humanos x INFORMACIÓN GEN x +

blob:http://localhost:4200/e3afe541-928e-45d0-97b1-d5a4413398ed

**Gobierno Autónomo de CHUQUISACA**

GOBIERNO AUTONOMO DEPARTAMENTAL DE CHUQUISACA  
DIRECCIÓN GENERAL DE RECURSOS HUMANOS

REF : INFORMACIÓN GENERAL  
FECHA : 5 DE JULIO DE 2018

**NÚMERO DE DOCUMENTO**  
1037671

**ESTADO CIVIL**  
VIUDO(A)

**TELEFONO FIJO**

**PAIS DE NACIMIENTO**  
BOLIVIA

**MUNICIPIO**  
SAN LUCAS

**NÚMERO DE DOCUMENTO EXPEDIDO**  
CHUQUISACA

**GENERO**  
MASCULINO

**TELEFONO MOVIL**  
71169433

**DEPARTAMENTO DE NACIMIENTO**  
CHUQUISACA

**CORREO ELECTRONICO**

**NOMBRE COMPLETO**  
FELIPE ARANCIBIA TANGO

**DIRECCIÓN**  
CALLE LLAWARWAKAJ N°150 BARRIO SICA SICA

**FECHA DE NACIMIENTO**  
04/11/1947

**PROVINCIA DE NACIMIENTO**  
NOR CINTI

supervisado por : HOSMAR ALMENDRAS PALGUERRA , TÉCNICO I

## ADAPTABLE EN LA WEB

DIRECCIÓN DE RRHH

**Gobierno Autónomo de CHUQUISACA**

\* USUARIO  
mi usuario...

\* CONTRASEÑA  
\*\*\*\*\*

Entrar

Bienvenido(a) al Sistema de Recursos Humanos, estamos felices de que seas parte de nuestro equipo de trabajo.  
**QUE TENGAS UN EXCELENTE DÍA.**

¿ Olvidaste tu contraseña ?

copyright © 2018 - alanwalter45

DIRECCIÓN DE RRHH

**INFORMACIÓN GENERAL**

\* CARNET DE IDENTIDAD  
1037671

\* CARNET EXPEDIDO  
CHUQUISACA

\* NOMBRES  
FELIPE

\* APELLIDO PATERNO  
ARANCIBIA

\* APELLIDO MATERNO  
TANGO

\* ESTADO CIVIL

\* GENERO  
MASCULINO

\* DIRECCIÓN

copyright © 2018 - alanwalter45

## PROTOTIPADO DE SOFTWARE

Durante el transcurso y fundamentación del análisis y marco de desempeño, se desarrollaron 23 prototipos de los distintos perfiles, tomando en cuenta que son 10 perfiles consideramos un promedio de 2.3 prototipos por perfil.

$$\text{TIEMPO} = (23/10) * (\text{tiempo estimado de trabajo})$$

Los prototipos fueron registrados usando GIT como tecnología para control de versiones.

## RECURSOS FACILITADOS – INFORMACIÓN

- Planilla de centralización en formato Excel con extensión XLSX.
- SAP – utilizado para las vacaciones.
- Files personales.
- Planilla de sueldos.

- Planilla de asistencia.
- Base de datos Access MDB.
- Otros documentos.

## TECNOLOGÍAS UTILIZADAS

Las tecnologías utilizadas para la elaboración del sistema se realizaron de acuerdo un análisis de alcance de software y predicción de soporte.

### ANGULAR V6

Angular es una plataforma y framework para construir aplicaciones en HTML y Typescript, Este implementa funcionalidades nativas y opciones junto con librerías Typescript que puedes importar en tus aplicaciones.

### HTML, CSS Y JAVASCRIPT

Una de las versiones que vino a revolucionar la web ha sido el lanzamiento de HTML5 que incluye mejoras en la manera de maquetar nuestros diseños dentro HTML por consecuencia la utilización de sus complementarios CSS3 encargado del estilo y JS encargado de funcionalidad, hablar de HTML5 sería conveniente hablar de estas 3 tecnologías.

- HTML es un lenguaje de marcado.
- CSS es un lenguaje de estilo.
- JavaScript el lenguaje de programación orientado a la web.

### TYPESCRIPT

Typescript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un supe conjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases. Anders Hejlsberg, diseñador de C# y creador de Delphi y Turbo Pascal, ha trabajado en el desarrollo de Typescript puede ser

usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (Node.js).

## **BOOTSTRAP**

Bootstrap es el framework más popular HTML, CSS, and JS para desarrollar proyectos responsivos enfocados en dispositivos móviles sobre la web.

## **NODEJS**

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.<sup>2</sup> Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla

## **EXPRESSJS**

Framework utilizado para manejar las peticiones realizadas a nuestra aplicación.

## **GULP**

Gulp es una herramienta para la automatización de tareas que consumen mucho tiempo en tu entorno de trabajo.

Se pueden disminuir las tareas repetitivas y crear un flujo de trabajo basado en la automatización de tareas.

- limpiar archivos de distribución generados en el anterior bundle
- ofuscación de código
- copiado de archivos a la distribución Docker



## MYSQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo,<sup>12</sup> y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

```
$ apt-get update
```

```
$ apt-get install mysql mysql-server
```

## FIREBIRD

Todos los sistemas necesitan la separación de datos o de estructuración de la información debido a que muchos procesos involucran portar la información de un lado a otro, y viendo que la red es interna se ve por conveniente utilizar una base de datos portable

```
$ apt-get update
```

```
$ apt-get install firebird3-server
```

```
$ isql-fb
```

```
$ CREATE DATABASE '/var/local/dbbird/recursoshumanos.fdb'
```

```
$ CREATE DATABASE '/var/local/dbbird/recursoshumanos.fdb' USER 'sysdba'
PASSWORD 'masterkey';
```

```
$ CONNECT '/var/local/dbbird/recursoshumanos.fdb' USER 'sysdba' PASSWORD
'masterkey';
```

## SQLITE

Es un sistema de gestión de [bases de datos relacional](#) compatible con [ACID](#), contenida en una relativamente pequeña (~275 [kiB](#))<sup>2</sup> [biblioteca](#) escrita en [C](#). SQLite es un proyecto de [dominio público](#)<sup>1</sup> creado por [D. Richard Hipp](#).

NOTA: es necesario Python instalado en el Sistema Operativo (WINDOWS o LINUX) para poder realizar la instalación.

También será necesario instalar Windows-Build-Tools

```
$npm install --global --production windows-build-tools
```

```
$npm install sqlite3 --save
```

## NATIVESCRIPT

NativeScript es una tecnología diseñada para construir aplicaciones cruzadas, aplicaciones nativas iOS y Android sin utilizar web views. Usa Angular, TypeScript o [avaScript](#) moderno para conseguir verdaderas UI nativas y desempeño mientras puedes reutilizar código en tus proyectos web. Obten el 100% de acceso a API's nativas via JavaScript y reuse paquetes desde npm, CocoaPods and Gradle. Open source y backed by Progress.

Instalación con npm

```
$ npm install --global nativescript
```

## ELECTRON JS

El framework Electron te permite escribir aplicaciones de escritorio cross-platform usando JavaScript, HTML and CSS. Está basado en [Node.js](#) y [Chromium](#) y es usado por el editor Atom y muchas otras aplicaciones.

## MOLECULAR JS

Framework que permite gestionar el desarrollo con micro-servicios , balanceo de carga , protocolo de comunicación y otros , MolecularJS te ayudará a orientas la arquitectura adecuada para la gestión de procesos.

## NODEMON

Herramienta que permitirá trabajar en un entorno de desarrollo y actualizar automáticamente en workflow de código NODE

```
$npm install nodemon --save
```

## PM2

Un paquete que te permitirá crear listas de aplicaciones y gestionar aplicaciones NODEJS

```
$npm install pm2 --global
```

## DOCKER

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de Virtualización a nivel de sistema operativo en Linux. Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que "contenedores" independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

## **GIT**

Es un sistema de control de versiones free y open source diseñado para manejar todo tipo de proyectos (grandes, pequeños) con alta eficiencia y velocidad.

## **JASMINE JS**

Es un framework para testing de código Javascript, está orientado a comportamientos. Este no depende de ningún otro framework. No requiere un DOM y este tiene una sintaxis limpia que te facilita escribir test.

## **KARMA JS**

Es un test Runner para aplicaciones de testeo, y es preferido por el equipo de Angular para este tipo de proyectos, te realizará la vida más fácil y tendrás una herramienta apta para el testing.

## **CARACTERÍSTICAS DEL SISTEMA**

### **AUTENTICACIÓN BASADA EN TOKENS**

Una de las nuevas tendencias en cuanto al desarrollo web moderno se refiere, es la autenticación por medio de Tokens y que nuestro backend sea un API RESTful sin información de estado, **stateless**.

El funcionamiento es el siguiente. El usuario se autentica en nuestra aplicación, bien con un par usuario/contraseña, o a través de un proveedor como puede ser Twitter, Facebook o Google por ejemplo. A partir de entonces, cada petición HTTP que haga el usuario va acompañada de un Token en la cabecera. Este Token no es más que una firma cifrada que permite a nuestro API identificar al usuario. Pero este Token no se almacena en el servidor, si no en el lado del cliente (por ejemplo en localStorage o sessionStorage) y el API es el que se encarga de descifrar ese Token y redirigir el flujo de la aplicación en un sentido u otro.

Como los **Tokens son almacenados en el lado del cliente**, no hay información de estado y la aplicación se vuelve totalmente escalable. Podemos usar el mismo API para diferentes aplicaciones (Web, Mobile, Android, iOS) solo debemos preocuparnos de enviar los datos en formato JSON y generar y descifrar tokens en la autenticación y posteriores peticiones HTTP a través de un middleware.

## TIEMPO REAL – SOCKET.IO

Un sistema en tiempo real (STR) es aquel sistema digital que interactúa activamente con un entorno con dinámica conocida en relación con sus entradas, salidas y restricciones temporales, para darle un correcto funcionamiento de acuerdo con los conceptos de predictibilidad, estabilidad, control y alcance

Los sistemas en tiempo real están presentes en nuestra vida diaria, prácticamente en todo lo que nos rodea: en los aviones, trenes y automóviles, en el televisor, la lavadora o el horno de microondas, en los teléfonos celulares y en las centrales telefónicas digitales. Son un elemento imprescindible para garantizar la generación, transmisión y distribución de la energía eléctrica y para asegurar la calidad y la seguridad de incontables procesos industriales.

La principal característica que distingue a los STR de otros tipos de sistemas es el tiempo de interacción. Sin embargo, antes de continuar es necesario aclarar el significado de las palabras tiempo y real.

- La palabra tiempo significa que el correcto funcionamiento de un sistema depende no sólo del resultado lógico que devuelve la computadora, también depende del tiempo en que se produce ese resultado.
- La palabra real quiere decir que la reacción de un sistema a eventos externos debe ocurrir durante su evolución. Como una consecuencia, el tiempo del sistema (tiempo interno) debe ser medido usando la misma escala con que se mide el tiempo del ambiente controlado (tiempo externo).

## **ARQUITECTURA BASADA EN COMPONENTES Y MICROSERVICIOS**

Orientaremos el proyecto a utilizar arquitecturas que permiten la agilidad en el desarrollo de prototipos, mockups y commits

### **1.- DISEÑO - ARQUITECTURA BASADA EN COMPONENTES**

Suelen utilizar patrones de diseño tales como MVC, MVP, MVVM, etc.

Por naturaleza la manera de organizar el proyecto por inercia de las tecnologías utilizadas como angular, Docker, node, etc, vemos conveniente no realizar cambios en la arquitectura del scaffolding para nuestro proyecto basándonos en componentes como unidad de trabajo

### **2.- EJECUCIÓN - ARQUITECTURA BASADA EN MICRO SERVICIOS**

Cuando el sistema está ejecutándose es bueno tener una arquitectura basada en consumo de recursos (servicios) y que realice más despedazable nuestra aplicación para tener más escalabilidad en cuestión de funcionalidad.

## **ARQUITECTURA DOCKER – CONTENEDORES**

### **1.- CONTAINER – MYSQL**

### **2.- CONTAINER – PHPMYADMIN**

### **3.- CONTAINER – NODE**

### **4.- CONTAINER – PM2**

### **5.- CONTAINER – UBUNTU**

Creando imagen nodejs (debian 8) que ejecutará la aplicación de recursos humanos, para ello utilizaremos un archivo de configuración llamado “dockerfile” donde especificamos las características de la imagen.

## Dockerfile

Construir container con dockerfile

```
$ docker build . -t node_rh:latest
```

rodar contenedor

```
$ docker run -it -p 3000:80 node_rh /bin/bash
```

## docker compose

Instalar docker compose

```
$ sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-  
compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
```

Permiso de ejecución

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

Averiguar versión

```
$ docker-compose --version
```

Construir la orquestación de servicios declarados y creación de imágenes

```
$ docker compose build
```

levantar los servicios declarados en docker-compose.yml

```
$ docker compose up -d
```

Utilizaremos los siguientes puertos para trabajar con la aplicación

**PUERTO 3000 - WEBAPP DE RECURSOS HUMANOS**

Accede a una aplicación web que le permite interactuar con la aplicación

Alan Walter Machuca Durex - [alanwalter45@gmail.com](mailto:alanwalter45@gmail.com)

## PUERTO 3006 - MYSQL

Accede al puerto para poder conectarte con un gestor de base de datos Ej. MySQL WorkBench 5.17

## PUERTO 3080 – PHPMYADMIN

Accede a phpmyadmin habilitado desde el navegador para poder gestionar la base de datos

-----  
3 instancias levantadas del sistema

15 conexiones habilitadas para las peticiones

Habilitar usuarios en MySQL del lado del servidor y dentro de Docker para dar permisos

```
$ create user 'root'@'172.18.0.3' identified by '@Alanwalter45'
```

## SISTEMA DE DESPLIEGUE – DOCKER

El llevar acabo el despliegue de aplicaciones se torna tedioso de acuerdo al tipo de proyecto que hayamos establecido, aunque por lo general el despliegue está más orientado a como establecer los parámetros necesarios para que nuestro sistema se ejecute y funcione con toda su arquitectura montada.

Para ello hemos montado una arquitectura basada en contenedores ya que debido a su portabilidad nos facilitaran la vida cuando queramos cambiar de servidor o simplemente instalar más nodos de cómputo.



## TEST UNITARIOS

En **programación**, una **prueba unitaria** es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo en **diseño estructurado** o en **diseño funcional** una función o un procedimiento, en **diseño orientado a objetos** una clase. Esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado. Además de verificar que el código hace lo que tiene que hacer, verificamos que sea correcto el nombre, los nombres y tipos de los parámetros, el tipo de lo que se devuelve, que si el estado inicial es válido entonces el estado final es válido

La idea es escribir casos de prueba para cada función no trivial o **método** en el módulo, de forma que cada caso sea independiente del resto. Luego, con las **Pruebas de Integración**, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

## CONFIGURACION DE LA APLICACIÓN

Durante la instalación del sistema es necesario estar modificando parámetros que serán de utilidad por motivo de seguridad o por comodidad a la hora de clonar más servidores en estaciones distintas.

Utilizaremos para el servidor NODEJS un archivo de configuración con extensión YAML  
Utilizaremos para el cliente HTML5 un archivo de configuración con extensión \*.js  
(NODE)

- YAML

- VANILLA NODE

## METODOLOGÍA DE DESARROLLO

Se vio por conveniente elegir una metodología de trabajo para el ciclo del proyecto y tomando en cuenta que solo se está trabajando con los requerimientos mínimos, además de compartir otras actividades o funciones, como oportunidad se optó por la **programación extrema**.

La programación extrema o eXtreme Programming (de ahora en adelante, XP) es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

## CONCLUSIONES

La implementación de una herramienta como la descrita en este documento, sería de mucha importancia para viabilizar el movimiento de operaciones rutinarias como la búsqueda de files, obtener listas de personal, etc.