



Informe técnico

Poyecto Final

Prof: Germán Rodríguez

Ederli, Florencia

DNI: 33.620.522

florencia.espino@gmail.com

Hotes, Emiliano

DNI: 32.300.181

es.hotes@gmail.com

Mitono, Laura

DNI: 29.192.942

mlauramitono@gmail.com

Sepúlveda, Florencia

DNI: 35.095.166

florenciasepulveda.26@gmail.com

DW6ATN - 2018

Acerca de Rádiko

Rádiko es una aplicación tanto web como mobile que le permite a huertas ponerse en contacto con clientes finales para venderles sus productos.

El cliente huerta carga sus productos en el sistema, recibe la información de las compras y puede ponerse en contacto con el cliente para pactar la entrega.

El cliente final tiene acceso a los productos que cargan las huertas y puede realizar una compra a través de la plataforma. También tiene la posibilidad de pagar a través de Mercado Pago y se puede poner en contacto con la huerta para pactar el envío.

www.radiko.com.ar



Índice de contenidos

- 5 Dominios
- 6 Servicios
- 7 Cuentas de correo
- 7 Librerías
- 7 Activación de módulos / paquetes
- 8 Tareas automatizadas de CRON

Dominios

Dominio Principal de proyecto

www.radiko.com.ar

Subdominios

dev.radiko.com.ar

Subdominio de desarrollo para trabajar en nuevas versiones de la aplicación antes de que sean llevadas a producción.

test.radiko.com.ar

Subdominio para hacer pruebas de usuario antes de lanzar nuevas versiones del sitio.

api.radiko.com.ar

Para la API REST.

admin.radiko.com.ar

Acceso de administrador al panel de control.

huertas.radiko.com.ar

Acceso para el usuario de tipo huerta.

mobile.radiko.com.ar

Para TESTEAR la app mobile.

Servicios

como administradores, con mayor permiso para la modificación de tablas.

SSH

- **Firewall:** remoto.

Ya que es un mecanismo de administración remoto de nuestro sistema, a través de claves públicas/privadas podemos acceder al servidor desde fuera para modificar lo que sea necesario.

Se cambiaría el puerto 22 a otro para aumentar la seguridad y evitar ataques externos.

- **Usuarios:** Tendríamos un usuario para cada uno de nosotros para poder acceder a través de ssh.

Por el momento perteneceríamos al grupo 'developers' o 'admins' y tendríamos sólo nosotros acceso a través de AllowUsers.

Además no se permitirán contraseñas vacías (PermitEmptyPasswords no) y se limitará el número de intentos de login fallido (MaxAuthTries 3).

FTP

- **Firewall:** remoto.

Primero se habilitarán los puertos 20 y 21 para FTP ya que iptables por default no los tiene configurados como Accept.

Al igual que ssh, necesitamos tener ac-

ceso remoto, en este caso para la administración y modificación de archivos.

- **Usuarios:** También habrá un usuario por integrante del grupo.

Servidor de correo

- **Firewall:** local.

El iptables para este servicio debe aceptar Output e Input en el puerto 25.

En este caso es local porque el servidor de email está en el mismo servidor y no en otro.

Además es local porque todo lo que sucede dentro del mail pasa dentro del servidor y no de forma externa (Administración de carpetas, emails, etiquetas, etc).

CRON

- **Firewall:** local.

Nadie externo tendrá acceso a CRON. Son acciones determinadas por nosotros en el servidor que se ejecutarán en un periodo predeterminado de tiempo (estas tareas se explicarán más adelante).

HTTPD

- **Firewall:** remoto.

Es remoto ya que cualquier usuario puede acceder a la web a través de los puertos 80 o 443.

MySQLd

- **Firewall:** local.

MySQLd solo puede tener acceso local por seguridad, ya que dejarlo abierto para usuarios remotos puede implicar posibles ataques.

- **Usuarios:**

- » *datamanager*: con permiso para modificar los valores de las tablas. En nuestro caso tiene permisos para update, select y create (ya que nosotros sólo hacemos borrado lógico y no físico, no necesitamos que tenga permisos para "delete"). Este usuario es el que va a usar el sitio para interactuar con la base de datos.
- » *root*: va a poder modificar las columnas de las tablas y los atributos de las filas (por ejemplo tipo de datos). Este usuario lo utilizamos nosotros

Cuentas de correo

hola@radiko.com.ar

Contacto general por cualquier consulta relacionada a la aplicación o el sitio.

huertas@radiko.com.ar

Contacto para huertas.

restaurantes@radiko.com.ar

Contacto para restaurantes.

pagos@radiko.com.ar

Cuenta que será utilizada para enviar mails en caso de que haya algún problema con el pago de una suscripción.

ayuda@radiko.com.ar

Soporte técnico.

Librerías

Laravel 5.6

<https://laravel.com/docs/5.6/installation>
(composer global require "laravel/installer")

jQuery v3.3.1

<https://code.jquery.com/jquery-3.3.1.min.js>

Bootstrap 4.1.3

<https://getbootstrap.com/docs/4.1/getting-started/download/>

Less

<http://lesscss.org/>
(npm install -g less)

Wow

<https://github.com/graingert/wow>

Animate.css

<https://github.com/daneden/animate.css>

Popper.js

<https://popper.js.org/>
(npm install popper.js --save)

Slick Carousel v 1.8.1

<https://github.com/kenwheeler/slick/archive/v1.8.1.zip>

Vue.js

<https://github.com/vuejs/vue>

Activación de módulos / paquetes

Openssl

Para aumentar la seguridad y estabilidad de Composer

Barryvdh/laravel-debugbar

Barra de depuración para laravel
(composer require barryvdh/laravel-debugbar)

Intervention/Image

Para manipular imágenes en laravel
(\$ php composer.phar require intervention/image)

Tareas automatizadas de CRON

Tarea: **Mail Newsletter**

Periodicidad: El día 3 de cada mes a las 10am.

Envío del newsletter mensual.

Se automatiza para que el sistema envíe el newsletter a la lista de distribución en intervalos de 5 minutos para evitar caer en SPAM.

Tarea: **Mail Usuario Inactivo**

Periodicidad: El día 1 de cada mes a las 10am.

Manda automáticamente un mail cuando el usuario no se conectó hace por lo menos 30 días.

Se automatiza para que el sistema checkee en la base los usuarios que deben recibir el mail y lo mande automáticamente.

Tarea: **Mail Falta de Pago**

Periodicidad: El día 5 de cada mes a las 3pm.

Manda un mail a todos los usuarios que no hayan efectuado el pago de su suscripción.

Se automatiza para que se checkee los usuarios que corresponden y el envío se haga de manera automática.

Tarea: **Desactivar Cuentas**

Periodicidad: El día 20 de cada mes a las 3am.

Desactiva todas las cuentas que figuren con falta de pago.

Se automatiza para no tener que llevar un

control manual de los pagos.

Tarea: **Backup Database**

Periodicidad: Todos los días a las 3am.

Realiza un backup de la base de datos.

Se automatiza para que se realice todos los días.

Tarea: **Borra Backup Database**

Periodicidad: Todos los días a las 5am.

Confirma que el último backup se haya realizado correctamente y borra el anterior.

Se automatiza para que se realice todos los días.

Tarea: **Checkeo de enlaces**

Periodicidad: El día 15 de cada mes a las 4am.

Se checkean todos los enlaces del sitio para corroborar que funcionen y se envíe un mail de confirmación al administrador.

Se automatiza para realizar un testeo mensual de manera automática.

Proyecto Final

Prof: Germán Rodríguez

Ederli, Florencia

Hotes, Emiliano

Mitono, María Laura

Sepúlveda, Florencia

DW6ATN - 2018