



UNIVERSIDAD TECNOLÓGICA ISRAEL

TEMA:

MANUAL DE INSTALACION DEL PROTOTIPO WEB/MÓVIL PARA EL
REGISTRO DE BITÁCORAS PARA EL ÁREA DE CAMPO EN LA FINCA
FLORÍCOLA VALTHOMIG S.A.

AUTOR:

GUANDINANGO GUALAPURO JAIME VINICIO

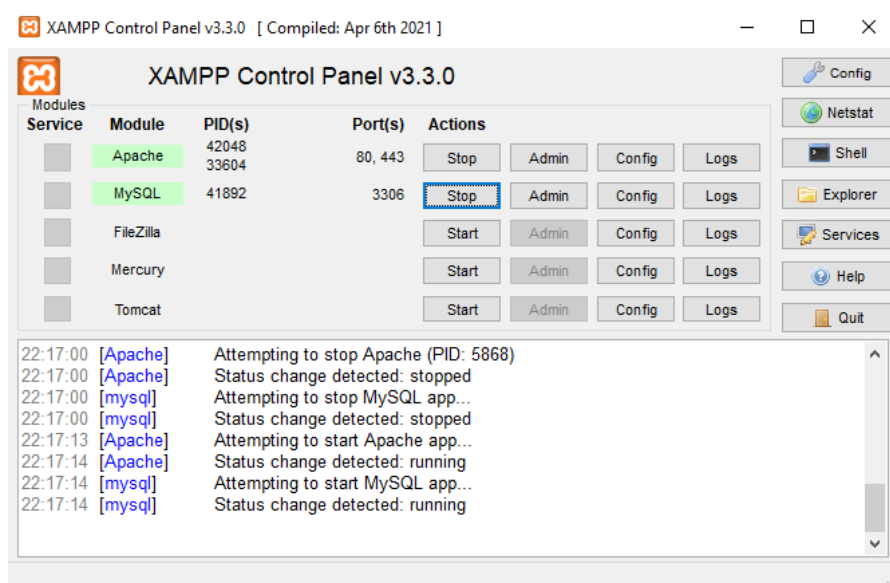
QUITO 2024

PASOS PARA LA INSTALACIÓN DEL PROTOTIPO WEB Y MOVIL

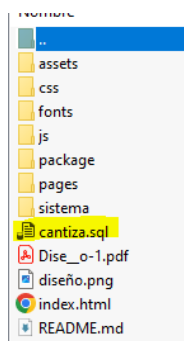
Arrancar el prototipo web de rendimientos

En esta parte del manual se revisarán los pasos a realizar para poder arrancar el prototipo web de rendimientos.

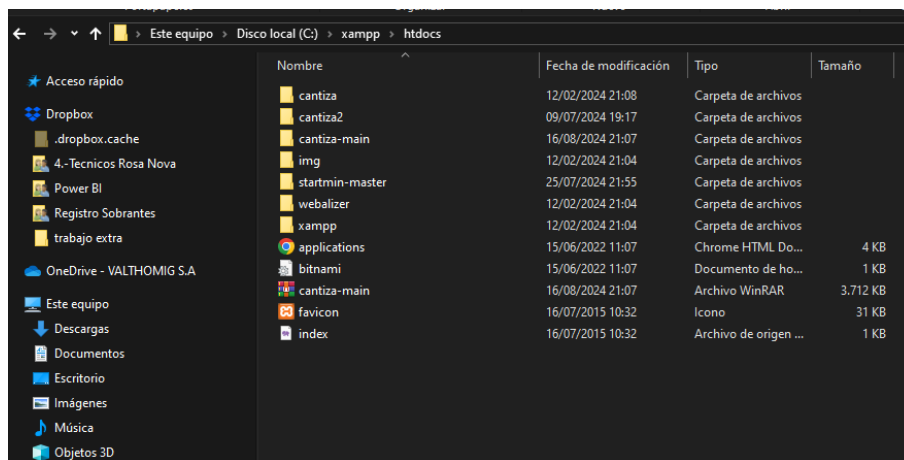
Primero levantamos los servicios para eso abrimos el programa Xampp y en su panel arrancamos los servicios del apache y MySQL, servicios que se necesitaran para poder arrancar localmente el prototipo.



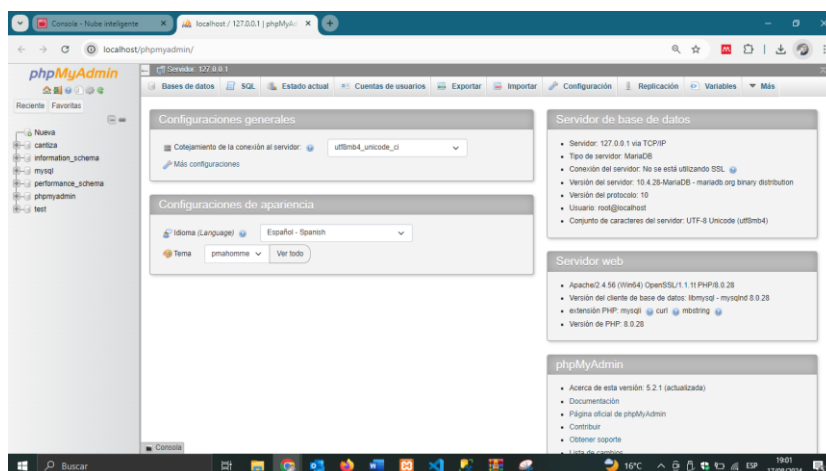
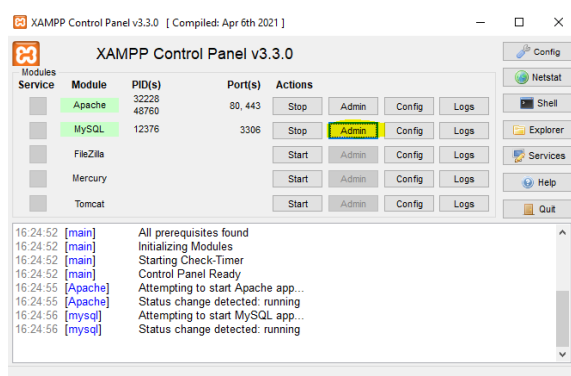
Descargamos el proyecto del github en el siguiente link en el cual esta almacenado el prototipo de rendimientos y descargamos para poder revisar el funcionamiento del mismo https://github.com/JaimeGuandinango/CP_GuandinangoJaime descargamos los archivos comprimidos el cual el archivo cantiza-main contiene el sistema, el scrip de base de datos llamado cantiza.sql y otras carpetas que son complementos del sistema



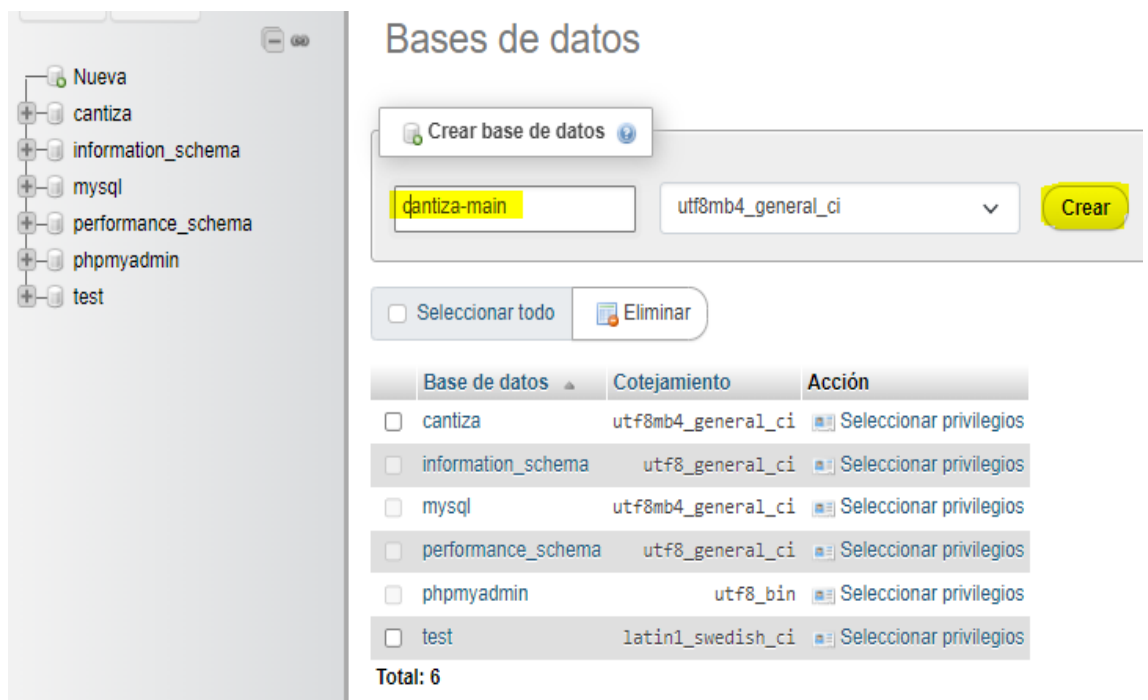
Copiamos toda la carpeta cantiza-main en la carpeta xampp\htdocs que está ubicado en el disco C de la maquina local para poder arrancar el sistema.



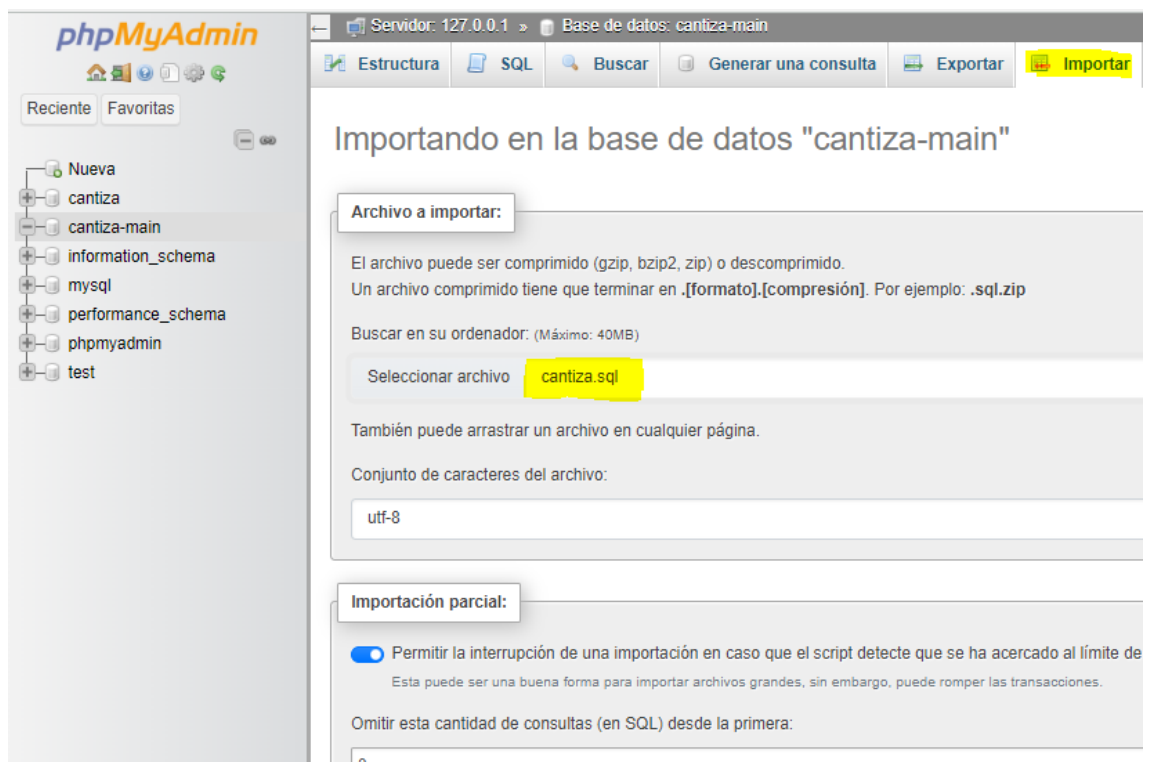
Luego nos dirigimos nuevamente al panel de xampp para importar la base de datos al servicio de MySQL y hacemos click en el botón que dice Admin coloreado de amarillo y nos dirigirá al phpMyAdmin.



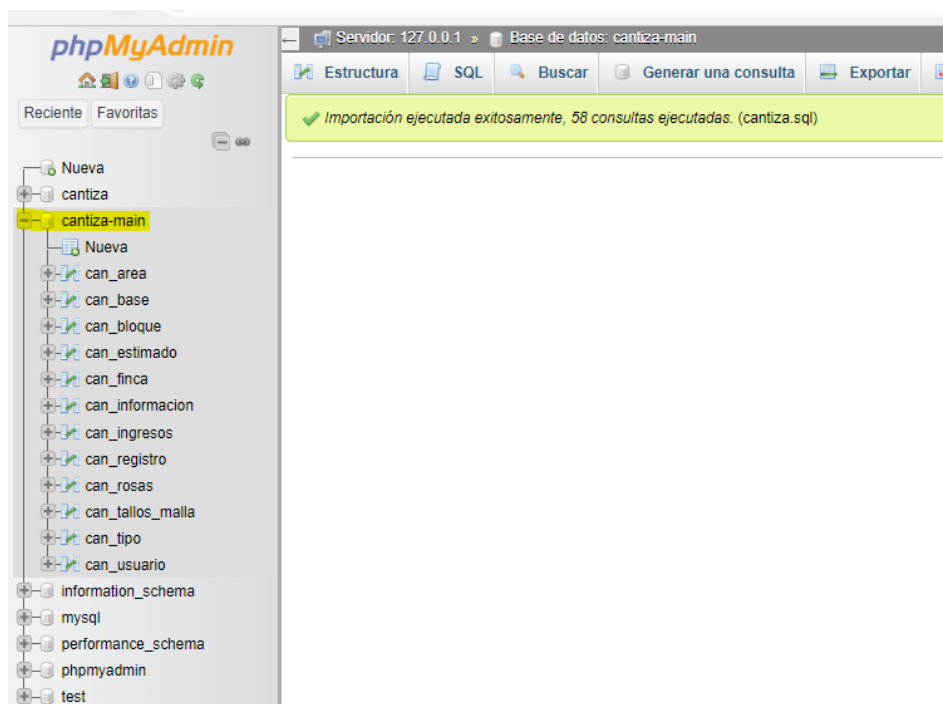
En esta parte nos ubicaremos en donde dice nueva para poder importar la base de datos, ponemos el nombre de la base de datos y hacemos click en crear.



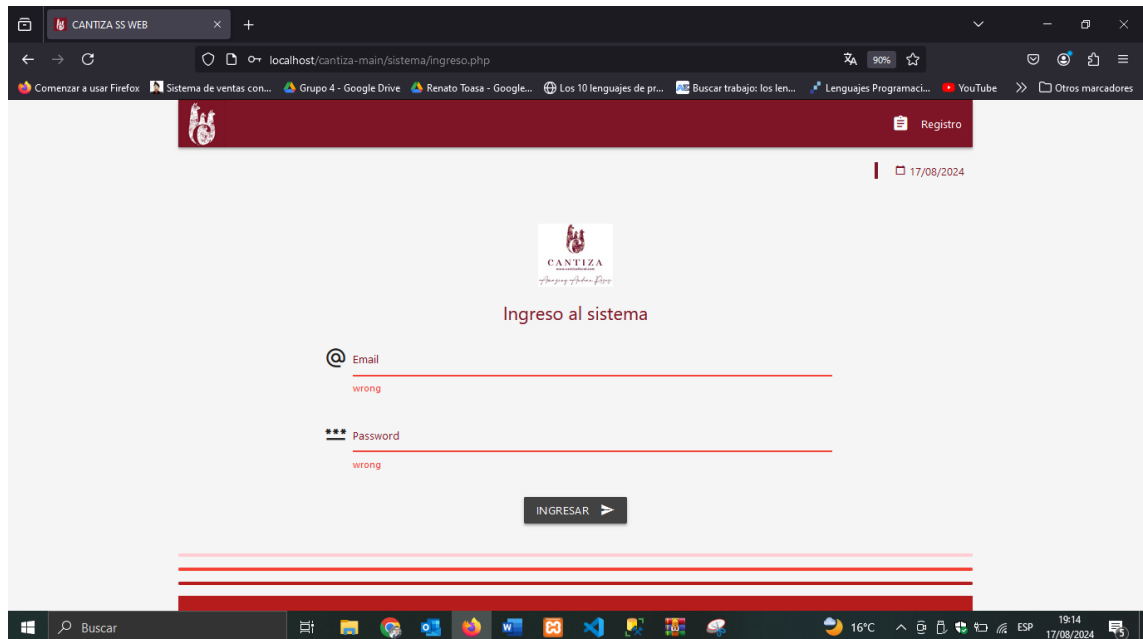
Luego seleccionamos el archivo que contiene el scrip de la base de datos y hacemos click en continuar para que la base de datos se cargue en el gestor de MySQL.



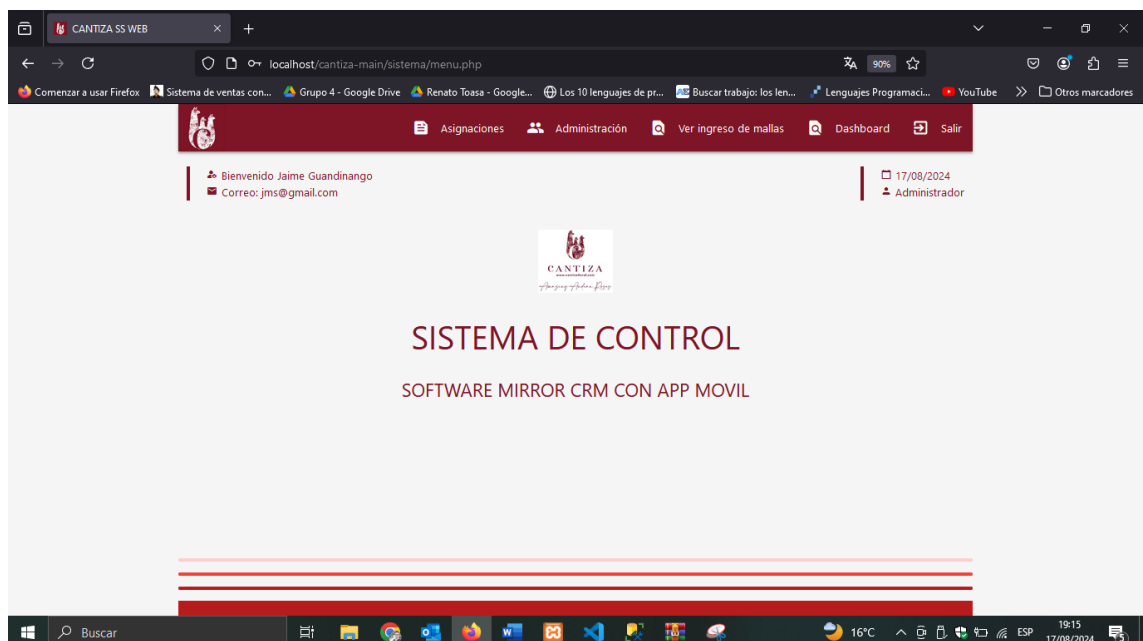
Una vez creada la base de datos podemos arrancar el proyecto web para revisar todas sus funciones que nos brinda el mismo.



Una vez terminado de cargar la base nos ubicamos en un navegador web de preferencia mozilla Firefox y digitamos lo siguiente para poder arrancar el prototipo de rendimientos, localhost/cantiza-main/sistema/ingreso.php el cual nos mostrará la interfaz de Login del sistema el cual solicitará el correo y la contraseña para poder ingresar.

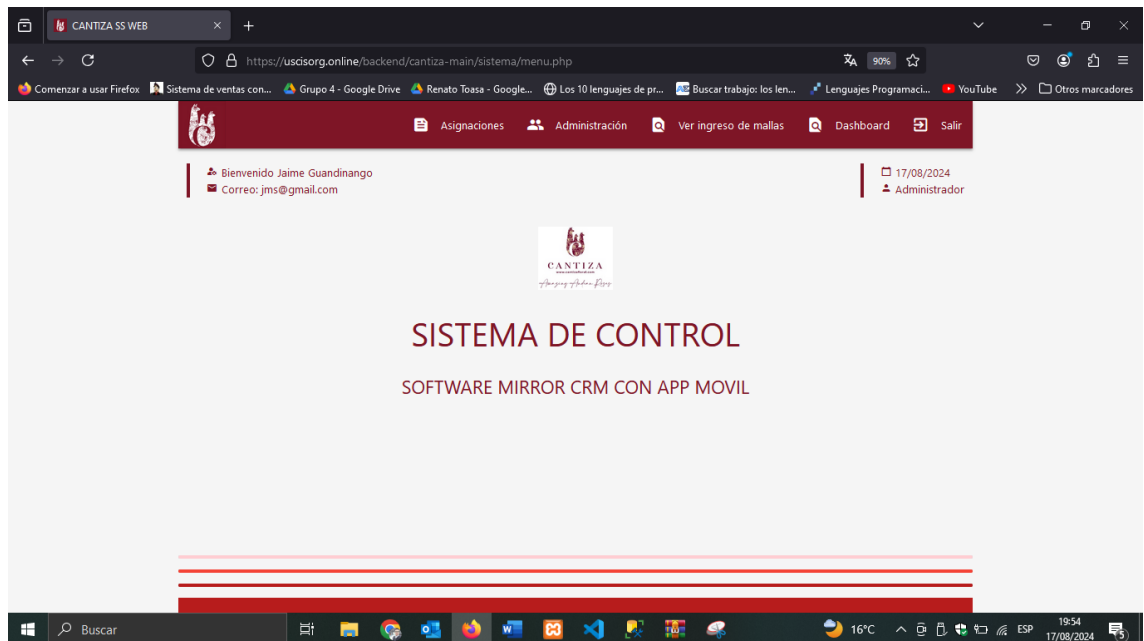
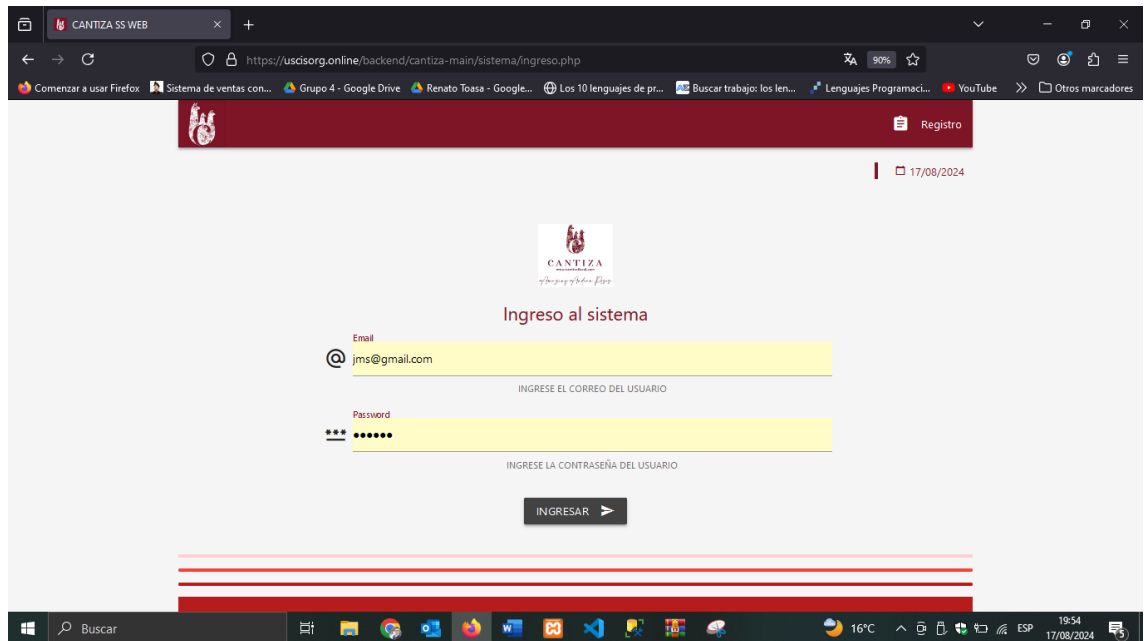


Si el inicio de sesión se realiza con éxito, se desplegará este menú de opciones con una barra superior dinámica y todo su contenido.



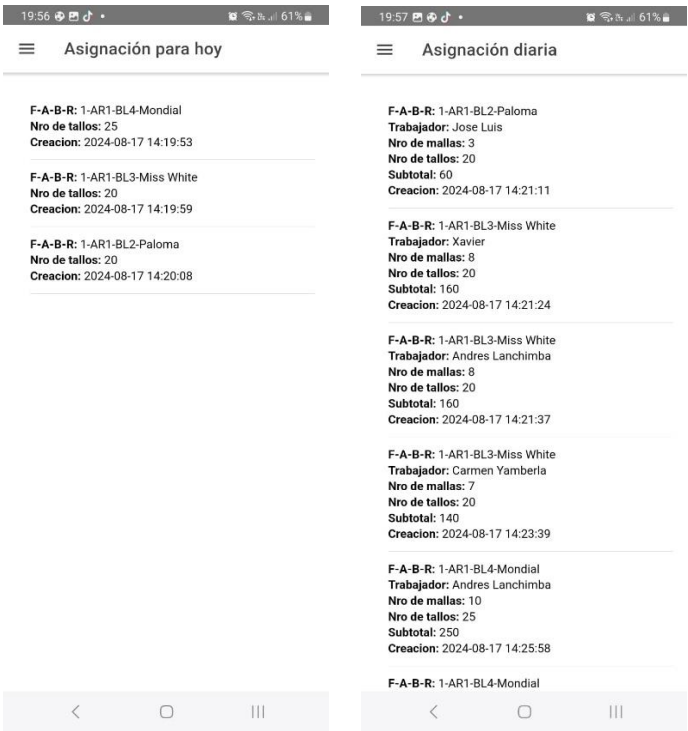
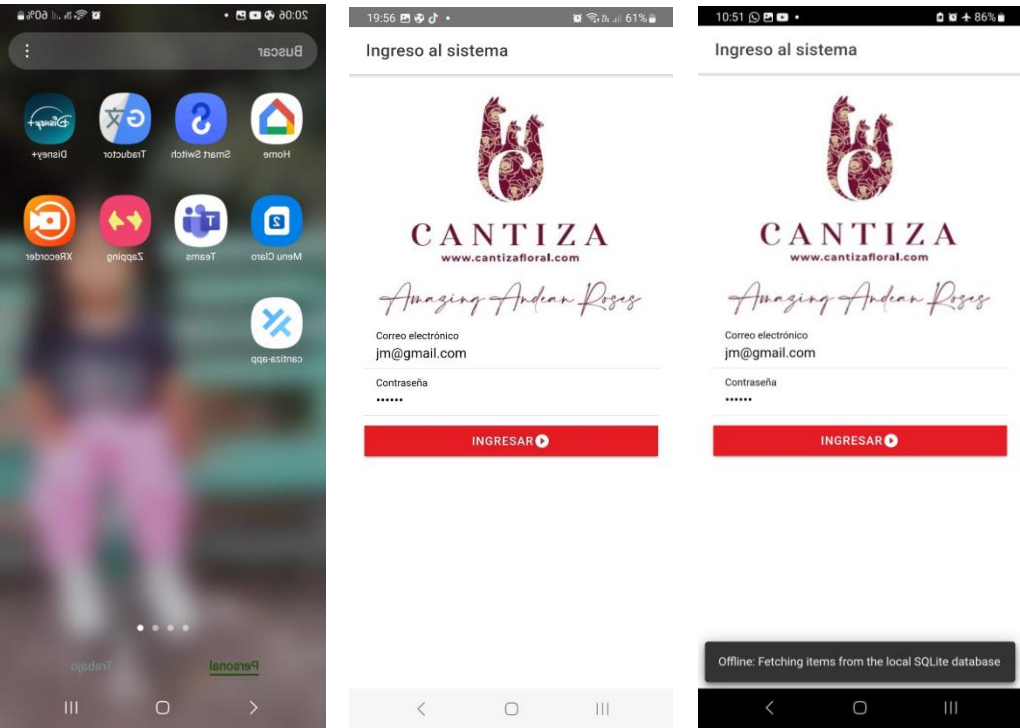
Arrancar el prototipo web publicada en un hosting en la nube.

Para arrancar el prototipo publicado en el hosting solo tendremos que ingresar al siguiente link <https://uscisgov.com/backend/cantiza-main/sistema/ingreso.php> el cual nos mostrará la ventana de Login en el cual tendremos que iniciar sesión con las credenciales brindadas por el administrador.



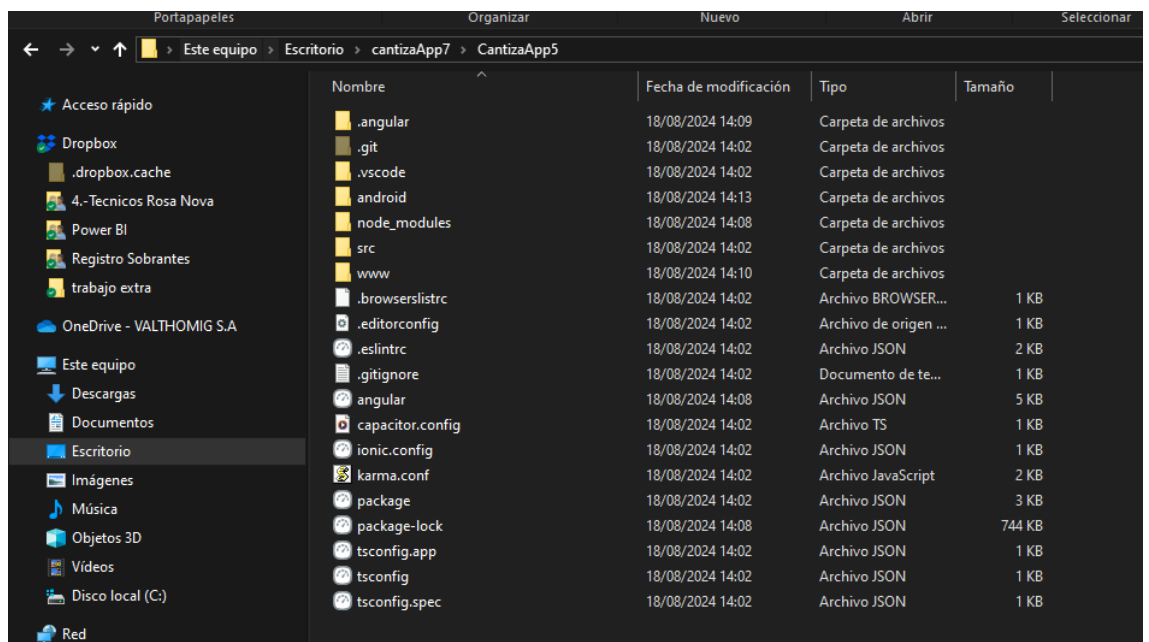
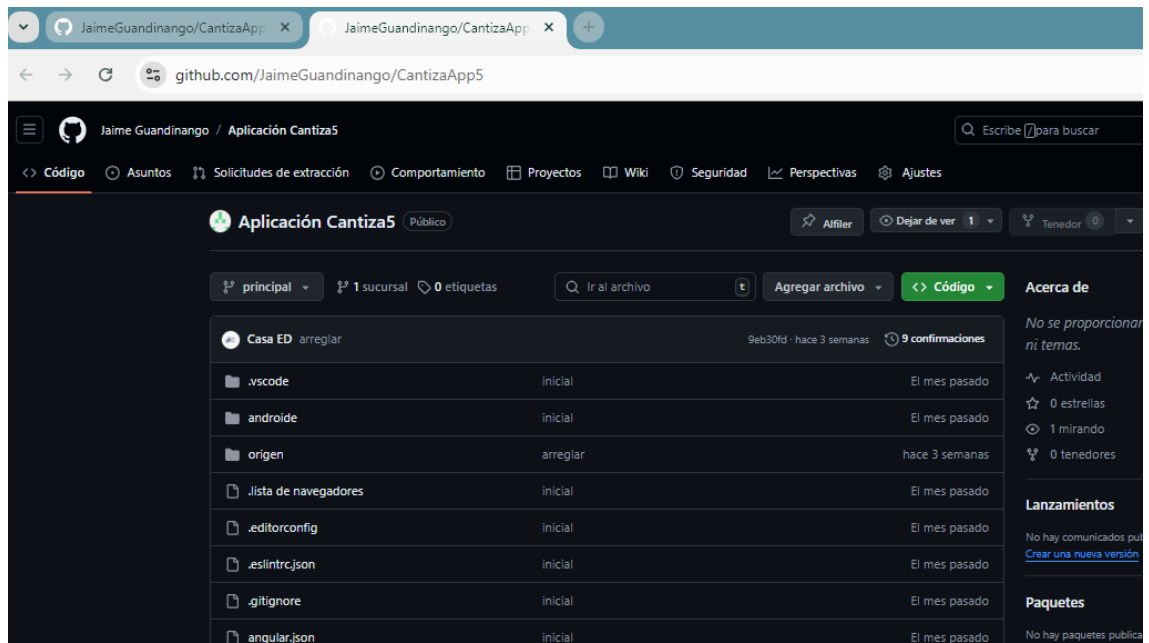
Arrancar la aplicación móvil en un dispositivo con sistema operativo Android

Para arrancar la aplicación móvil, solo bastará con descargar la apk del github e instalar en cualquier dispositivo con Android como sistema operativo de arranque y se podrá acceder al funcionamiento de la app móvil en modo online, como en modo off line, además del interfaz de la asignación diaria y el listado de ingreso de mallas.

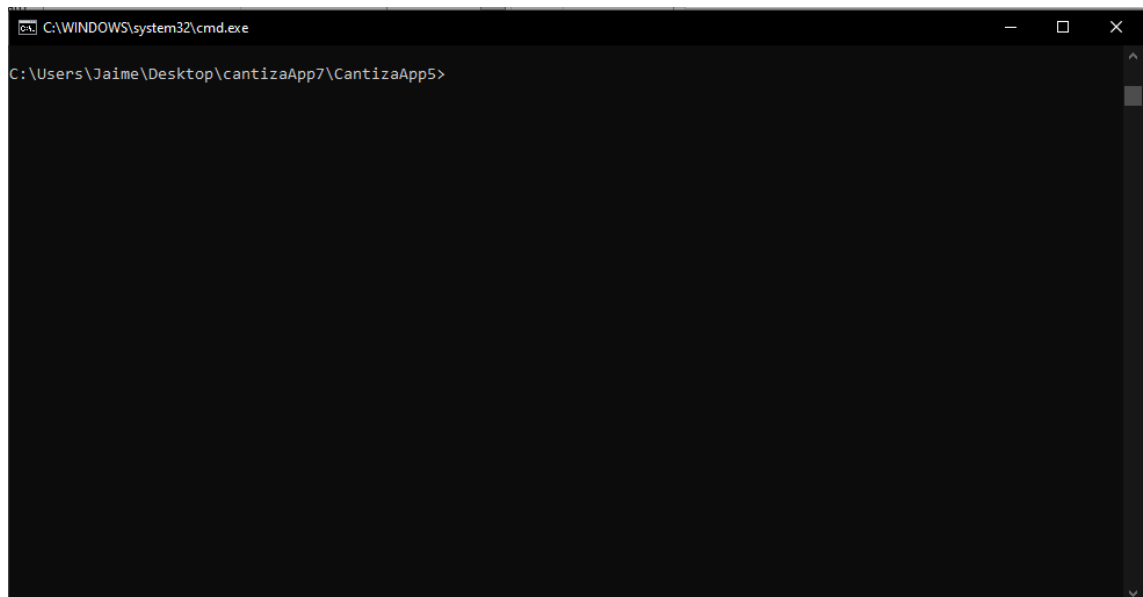


Como arrancar la app online de forma local

Para arrancar la app de forma local se deberá descarga el proyecto del github <https://github.com/JaimeGuandinango/CantizaApp5> y ubicarlo en una carpeta en la maquina local.

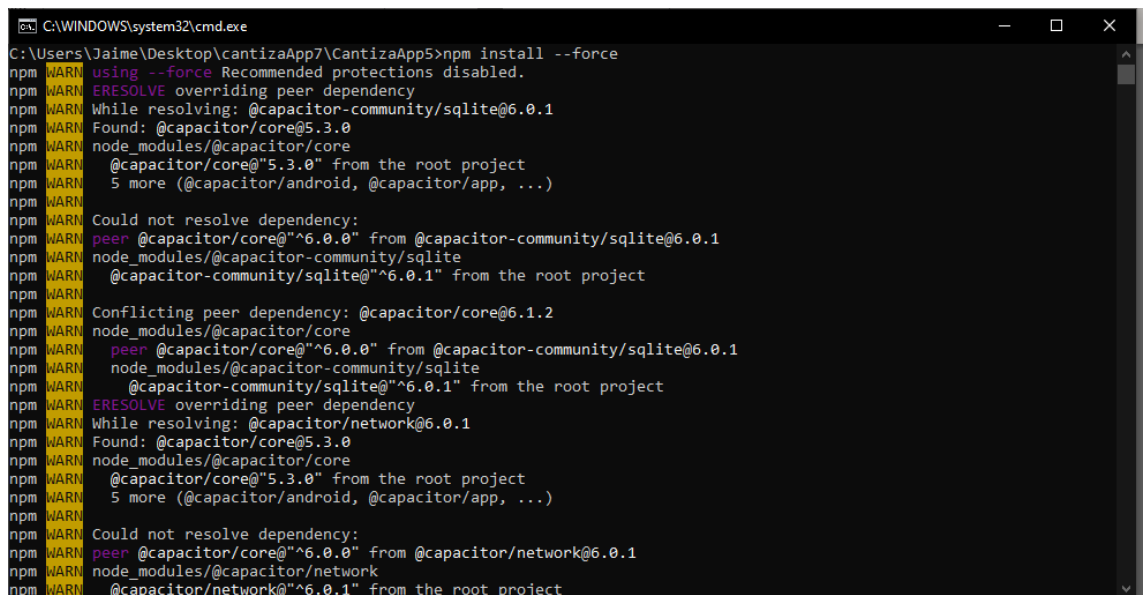


Luego nos ubicaremos en el terminal de la maquina o también conocido como el CMD y nos ubicamos en la carpeta del prototipo móvil.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Jaime\Desktop\cantizaApp7\CantizaApp5>
```

Una vez ubicados en el prototipo móvil instalaremos el npm que es un recurso que usa localmente la aplicación para poder arrancar con el comando `npm install --force` le damos un enter y procederá con la instalación.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Jaime\Desktop\cantizaApp7\CantizaApp5>npm install --force
npm WARN using --force Recommended protections disabled.
npm WARN ERESOLVE overriding peer dependency
npm WARN While resolving: @capacitor-community/sqlite@6.0.1
npm WARN Found: @capacitor/core@5.3.0
npm WARN node_modules/@capacitor/core
npm WARN   @capacitor/core@"5.3.0" from the root project
npm WARN   5 more (@capacitor/android, @capacitor/app, ...)
npm WARN Could not resolve dependency:
npm WARN peer @capacitor/core@"^6.0.0" from @capacitor-community/sqlite@6.0.1
npm WARN node_modules/@capacitor-community/sqlite
npm WARN   @capacitor-community/sqlite@"^6.0.1" from the root project
npm WARN Conflicting peer dependency: @capacitor/core@6.1.2
npm WARN node_modules/@capacitor/core
npm WARN   peer @capacitor/core@"^6.0.0" from @capacitor-community/sqlite@6.0.1
npm WARN   node_modules/@capacitor-community/sqlite
npm WARN     @capacitor-community/sqlite@"^6.0.1" from the root project
npm WARN ERESOLVE overriding peer dependency
npm WARN While resolving: @capacitor/network@6.0.1
npm WARN Found: @capacitor/core@5.3.0
npm WARN node_modules/@capacitor/core
npm WARN   @capacitor/core@"5.3.0" from the root project
npm WARN   5 more (@capacitor/android, @capacitor/app, ...)
npm WARN Could not resolve dependency:
npm WARN peer @capacitor/core@"^6.0.0" from @capacitor/network@6.0.1
npm WARN node_modules/@capacitor/network
npm WARN   @capacitor/network@"^6.0.1" from the root project
```

```
npm WARN Could not resolve dependency:
npm WARN peer rxjs@"^5.5.0 || ^6.5.0" from @ionic-native/core@5.36.0
npm WARN node_modules/@ionic-native/core
npm WARN peer @ionic-native/core@"^5.1.0" from @ionic-native/sqlite@5.36.0
npm WARN node_modules/@ionic-native/sqlite
npm WARN Conflicting peer dependency: rxjs@6.6.7
npm WARN node_modules/rxjs
npm WARN peer rxjs@"^5.5.0 || ^6.5.0" from @ionic-native/core@5.36.0
npm WARN node_modules/@ionic-native/core
npm WARN peer @ionic-native/core@"^5.1.0" from @ionic-native/sqlite@5.36.0
npm WARN node_modules/@ionic-native/sqlite
npm WARN RESOLVE overriding peer dependency
npm WARN While resolving: @ionic-native/sqlite@5.36.0
npm WARN Found: rxjs@7.8.0
npm WARN node_modules/rxjs
npm WARN rxjs@"7.8.0" from the root project
npm WARN 6 more (@angular/common, @angular/core, @angular/forms, ...)
npm WARN Could not resolve dependency:
npm WARN peer rxjs@"^5.5.0 || ^6.5.0" from @ionic-native/sqlite@5.36.0
npm WARN node_modules/@ionic-native/sqlite
npm WARN @ionic-native/sqlite@"^5.36.0" from the root project
npm WARN Conflicting peer dependency: rxjs@6.6.7
npm WARN node_modules/rxjs
npm WARN peer rxjs@"^5.5.0 || ^6.5.0" from @ionic-native/sqlite@5.36.0
npm WARN node_modules/@ionic-native/sqlite
npm WARN @ionic-native/sqlite@"^5.36.0" from the root project
[ ] - reify:fsevents: timing reifyNode:node_modules/nice-napi Completed in 13ms
```

Una vez instalado el npm digitaremos ionic serve dado que usa IONIC como herramienta tecnológica para su construcción porque nos brinda una escalabilidad impresionante para futuras mejoras a la aplicación móvil y teniendo un estándar en el mercado, dando posibilidades de realizar aplicaciones para Android como también para IOS.

```
C:\WINDOWS\system32\cmd.exe
[info] Opening Android project at: android.

C:\Users\Jaime\Desktop\cantizaApp7\CantizaApp5>ionic serve
> ng.cmd run app:serve --host=localhost --port=8100
```

```
Windows PowerShell
[ng] node_modules_ionic_core_dist_esm_ion-avatar_3_entry_js.js | 8.39 kB | -
[ng] node_modules_ionic_core_dist_esm_index-0037f985_js.js | 8.28 kB | index-003
7f985-js | 6.34 kB | -
[ng] node_modules_ionic_core_dist_esm_ion-ripple-effect_entry_js.js | 6.25 kB | -
[ng] node_modules_ionic_core_dist_esm_ion-img_entry_js.js | 4.63 kB | -
[ng] node_modules_ionic_core_dist_esm_ion-text_entry_js.js | 4.20 kB | -
[ng] node_modules_capacitor_network_dist_esm_web_js.js | 3.38 kB | web
[ng] node_modules_ionic_core_dist_esm_ion-backdrop_entry_js.js | 3.24 kB | -
[ng] node_modules_ionic_core_dist_esm_status-tap-db08b2e9_js.js | 2.92 kB | status-ta
p-db08b2e9-js
[ng] Build at: 2024-08-17T02:52:08.875Z - Hash: 48498110a4ebfea9 - Time: 39249ms
[ng] ✓ Compiled successfully.

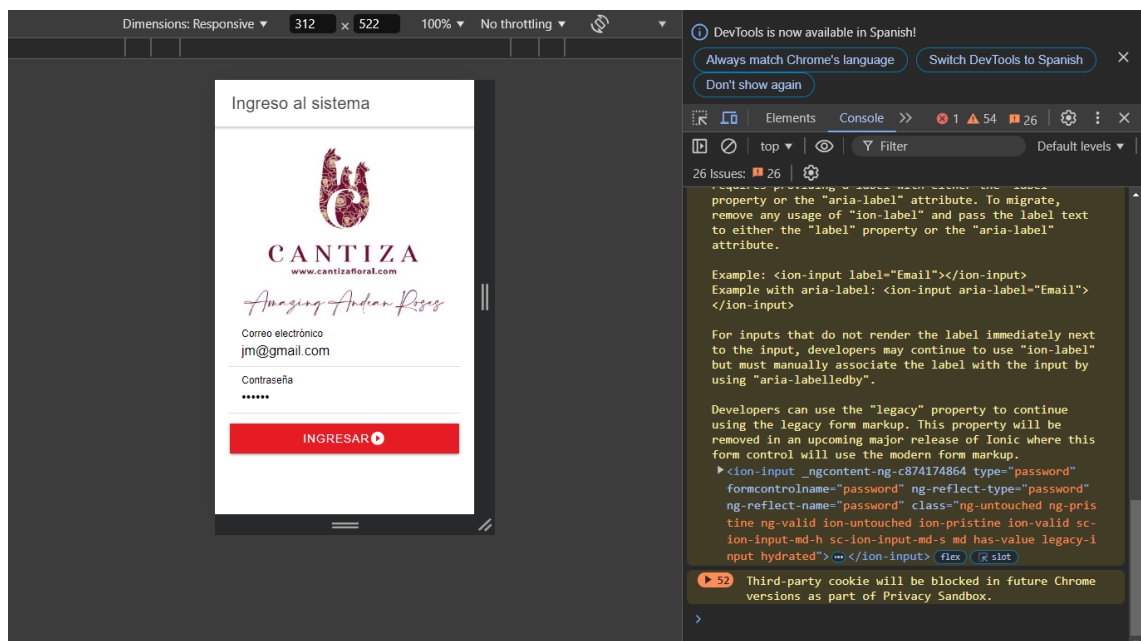
[INFO] Development server running!

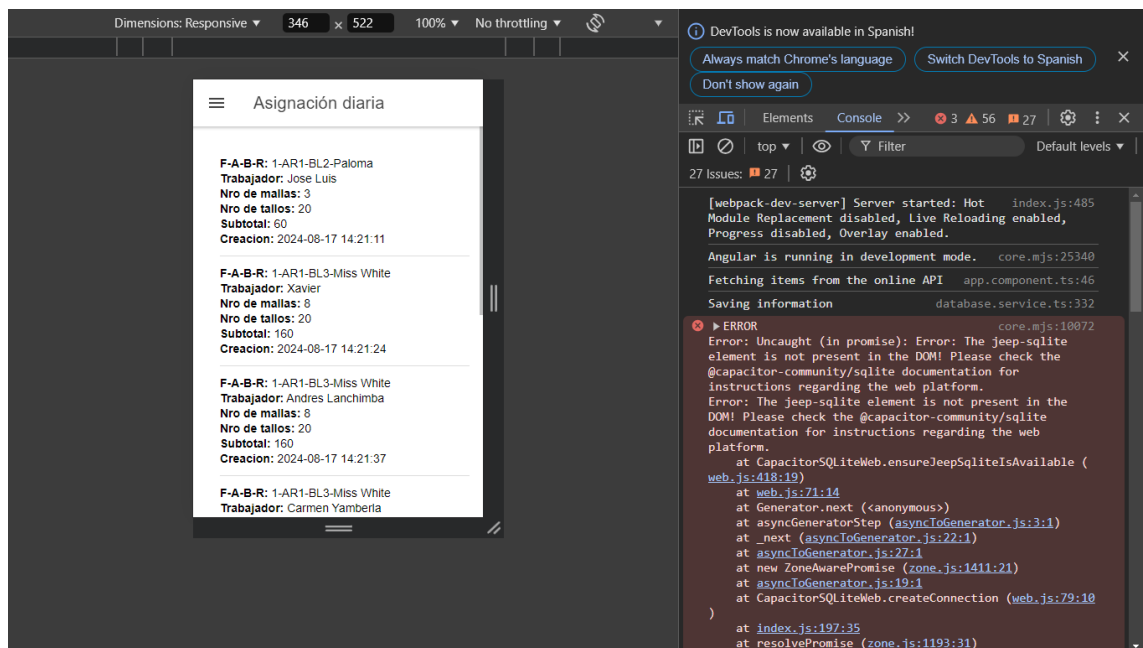
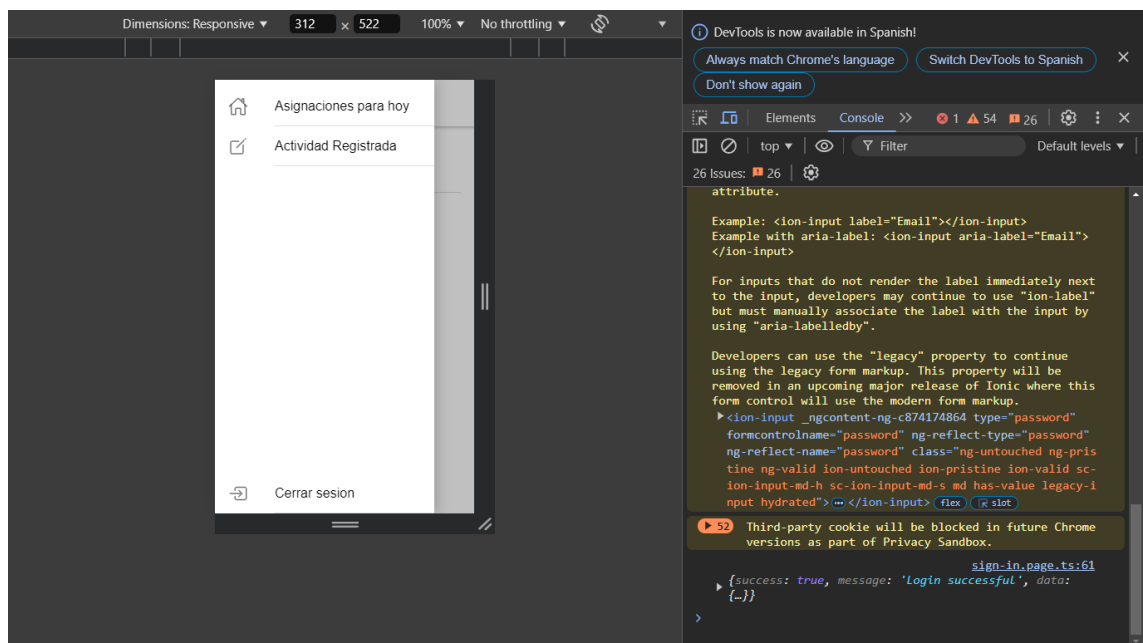
Local: http://localhost:8100

Use Ctrl+C to quit this process

[INFO] Browser window opened to http://localhost:8100!
```

Cuando termine de arrancar la apk tendremos la interfaz de la aplicación funcionando correctamente de forma online con todas sus funcionalidades.





Credenciales para la base de datos local	Credenciales para la base de datos en la nube
host="localhost"	<code>\$host = "sql10.freemsqldatabase.com"</code>
user="root"	<code>\$user = "sql10727160"</code>
clave=""	<code>\$clave = "XENqatLIXJ"</code>
bd="cantiza"	<code>\$bd = "sql10727160"</code>

Credenciales los cuales serán cambiados en el archivo de conexión según sea el caso el archivo que accederemos en la carpeta cantiza-main/sistema/config/link.php.

Nota: la aplicación web funciona con la base de datos en la nube porque usa recursos REST que necesariamente necesita estar alojado en una nube para realizar la sincronización automática cuando registre datos de forma online y offline.

En cuanto al prototipo web se usará las credenciales de base de datos locales solamente para revisar su funcionamiento local como se indica ya para revisar la funcionalidad con la aplicación respectivamente es necesario que la base de datos apunte hacia las credenciales que están publicadas en la nube.

Credenciales para revisar el sistema usuario administrador

User: jms@gmail.com

Pass: 123456

Credenciales para realizar la asignación diaria como usuario supervisor

User: ps@gmail.com

Pass: 123456

Credenciales para registrar los datos de mallas como usuario cochero

User: jm@gmail.com

Pass: 123456

En caso de poner el sistema en producción se cambiarán las claves por seguridad mientras tanto son solo pruebas realizadas con éxito.