

Una Aplicación Web Progresiva para la Gestión de Asistencia

Juan Fernández Sosa ¹, Fernando Tesone ¹, Marcelo Naiouf ¹, Marcos Boracchia ¹, Pablo Thomas ¹, Patricia Pesado ¹

¹ Instituto de Investigación en Informática LIDI (III-LIDI). Facultad de Informática – Universidad Nacional de La Plata, La Plata, Argentina.
Centro Asociado a la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)
{jfernandez, ftesone, mnaiouf, marcosb, pthomas, ppesado }@lidi.info.unlp.edu.ar

Resumen

La pandemia que aún está presente ha impuesto severas restricciones en la vida cotidiana. El ámbito educativo no ha sido eximido de tal situación. Este año, paulatinamente se están recuperando hábitos que en los últimos dos años se habían perdido.

En este trabajo se presenta el desarrollo de una aplicación web progresiva (PWA, su sigla en inglés) para la gestión de la presencialidad durante el curso de ingreso 2022 en la Facultad de Informática de la Universidad Nacional de La Plata.

Palabras clave: PWA - Desarrollo de aplicaciones móviles - Desarrollo de aplicaciones móviles multiplataforma

Introducción

La pandemia de COVID-19 ha tenido un gran impacto a nivel mundial en aspectos sociales, económicos, culturales y educativos. Haciendo hincapié en este último, las instituciones educativas en sus diferentes órdenes (primarias, secundarias, universitarias de grado y postgrado) se vieron obligadas a adoptar nuevas habilidades y competencias para educar a distancia. Esta metodología se extendería por un tiempo incierto e indeterminado por lo que se debió hacer uso de la tecnología para montar plataformas de enseñanza

remota. Este impulso por la adopción de nuevas tecnologías en el ámbito de la educación, debe sostenerse en esta nueva etapa de la pandemia, con la vuelta a la presencialidad.

Gradualmente, se espera volver a una etapa de normalidad, donde la presencialidad en las clases sea plena. Mientras tanto, se deben seguir y adoptar protocolos para continuar con los cuidados de la salud de la comunidad. Diferentes universidades presentaron sus protocolos para el regreso a la actividad académica presencial. En el caso de la Universidad Nacional de La Plata (UNLP), la Disposición 176/21 establece en su artículo primero, que para la participación en cualquier actividad presencial se debe acreditar estar vacunado con la primera dosis de alguna de las vacunas autorizadas destinadas a generar inmunidad contra el COVID-19. En el caso de la Facultad de Informática de la UNLP, el control del cumplimiento de esta restricción se lleva a cabo por medio de una declaración jurada de salud.

Una de las primeras actividades académicas presenciales que se dictó en la Facultad de Informática fue el curso de ingreso. Dentro de las condiciones para ingresar a la facultad se requiere contar con al menos el 80% de asistencia presencial a los diferentes módulos disciplinares que se

dictan. Por tal motivo, se lleva a cabo un registro y control de asistencia de cada estudiante ingresante. La Facultad desde hace algunos años cuenta con un novedoso sistema de toma de asistencia por medio de la detección de huella dactilar. Sin embargo, el contexto pandémico amerita un cambio de sistema por uno de tipo *contactless*. Es decir, un sistema con el que se pueda interactuar sin necesidad de contacto físico.

Este trabajo tiene como objetivo presentar la experiencia del desarrollo de una solución tecnológica para la toma de asistencia en el curso de ingreso, y el control de la declaración jurada de salud para ingresar a la Facultad de Informática de la UNLP durante el curso de ingreso 2022.

QR: Código de respuesta rápida

Los códigos QR (Quick Response Code, en inglés) son gráficos bidimensionales que permiten almacenar información, que luego puede ser consultada con la lectura a través de un dispositivo móvil.

Existen múltiples sistemas que hacen uso de esta popular tecnología y esto se debe en parte a los beneficios que se obtienen de su utilización, entre los cuales se pueden destacar la rápida lectura, la lectura omnidireccional, el tamaño pequeño, los datos pueden ser leídos aunque esté dañado hasta un 30% del código, son fácil de generar, entre otras características [1].

En particular, esta tecnología tiene una gran adopción en la Facultad de Informática de la UNLP como puede verse en la figura 1. Al recorrer los pasillos de la unidad académica, pueden observarse códigos QR en la puerta de las aulas, los cuales pueden ser leídos utilizando la aplicación de software para dispositivos móviles (app) propia de la facultad [2]. También se puede encontrar esta tecnología

en las carteleras físicas mostrando información sobre las carreras de grado y postgrado, asignaturas, carteleras digitales, etc.



Fig 1. Ejemplos de códigos QR disponibles en la Facultad de Informática UNLP.

Esta tecnología que resulta familiar para docentes, no docentes y estudiantes en

el ámbito de la facultad, fue la escogida como alternativa al sistema de detección de huellas dactilares para tomar asistencia durante el ingreso presencial 2022.

A cada estudiante ingresante se le generó un código QR con información propia de cada uno de ellos (DNI, nombre y apellido, entre otros). Dicha información, la cual se encuentra contenida como texto plano en los códigos QR, se estructura de la siguiente manera para facilitar su procesamiento luego de ser leída:

DNI XXXXXXXX APELLIDO, NOMBRE NUMERO CARRERA - NOMBRE CARRERA INGRESANTE 2022 CÓDIGO DE INGRESANTE
--

Enfoques de desarrollo de aplicaciones multiplataforma móviles

Existen dos grandes estrategias para desarrollar una aplicación de software para dispositivos móviles: la estrategia nativa y la estrategia multiplataforma [3]. La primera de ellas consiste en desarrollar una aplicación específica para cada una de las plataformas o sistemas operativos en las que se desea distribuir. Para ello se deben emplear los lenguajes de programación y las herramientas de desarrollo (SDK) provistos por cada una de las plataformas. Se tendrá entonces por cada plataforma, un proyecto diferente.

En contrapartida al desarrollo nativo, ha surgido la estrategia multiplataforma con el objetivo de desarrollar un solo proyecto, con un único código fuente que será compilado y ejecutado en diferentes plataformas. Este enfoque de desarrollo puede llevarse a cabo empleando una gran diversidad de tecnologías, lo que da origen a

una clasificación de las estrategias de desarrollo multiplataforma en: aplicaciones web móviles, híbridas, interpretadas y generadas por compilación cruzada [4].

Las aplicaciones web móviles se ejecutan en un navegador y se desarrollan con tecnologías web conocidas tales como HTML, CSS y JavaScript. No necesitan adecuarse a ningún entorno operativo. En los últimos años ha surgido un nuevo concepto denominado Aplicaciones Web Progresivas (PWA por sus siglas en inglés). Una PWA es una aplicación web que utiliza las últimas tecnologías disponibles en los navegadores para ofrecer en dispositivos móviles una experiencia lo más parecida posible a la de una aplicación nativa [5]. Entre dichas tecnologías se encuentran los *Service Workers* que se encargan de realizar tareas en segundo plano y el Manifiesto de la Aplicación Web que permite configurar cómo mostrar la aplicación en dispositivos móviles.

La instalación de una PWA no está limitada sólo a dispositivos móviles, sino que se extiende también a computadoras de escritorio o dispositivos móviles con navegadores web compatibles con esta tecnología. Esto posibilita un único desarrollo de software independiente del tipo de dispositivo donde se utilizará. Dicha característica coloca a las PWA por sobre el resto de los enfoques de desarrollo multiplataforma [6].

Diseño y desarrollo del sistema

Requerimientos funcionales

El sistema desarrollado debía contar con dos funcionalidades principales. La primera de ellas, permitir la lectura de los códigos QR generados para los estudiantes y de ese modo registrar su asistencia a los módulos del curso de ingreso.

La Facultad de Informática cuenta con un software especializado para la gestión del curso de ingreso [7]. Dicho sistema, entre otras funcionalidades administrativas, permite el almacenamiento de la asistencia de los estudiantes, sus notas, etc. La asistencia en este sistema de gestión se ingresa de a un estudiante por vez de manera manual. Se aprovechó la oportunidad de tener que desarrollar una nueva solución tecnológica para la toma de asistencia, para que la sincronización de los registros en el sistema de gestión, se realicen de manera automatizada mediante una API Rest desarrollada específicamente para este fin.

La segunda funcionalidad desarrollada consiste en poder leer los mismos códigos QR de los estudiantes pero en este caso para acreditar si el estudiante realizó la declaración jurada de salud. Ambas funcionalidades se detallarán en las siguientes secciones.

Requerimientos no funcionales

Resulta especialmente interesante resaltar los requerimientos no funcionales que el sistema debe satisfacer, ya que la estrategia de desarrollo del software se eligió a partir del análisis y evaluación de cada uno de ellos.

El primer requerimiento no funcional a considerar es la *portabilidad*, el software debe poder ejecutarse en cualquier dispositivo móvil que cuente con una cámara fotográfica y un navegador web. Del universo de dispositivos móviles, aquellos a los que se busca dar soporte son smartphones (independientemente de su sistema operativo), laptops y tablets.

Por otra parte, se requiere que la operatividad del sistema no se encuentre sujeta a contar una conexión estable y

permanente a internet, es decir que la aplicación sea *resiliente* en lo relativo a conectividad. Este tipo de diseño se lo conoce como soluciones “offline-first” [8]. Las aplicaciones que utilizan este tipo de diseño permiten procesar y almacenar datos de manera local para que luego sean sincronizados de manera automática o por acción del usuario cuando se recupera la conectividad.

Además, el sistema debe ser de *fácil mantenimiento*. Esto significa que los tiempos de corrección de errores y despliegue del aplicativo deben ser cortos. Se debe considerar que en el desarrollo de aplicaciones de software para dispositivos móviles, se suelen distribuir las apps en las tiendas de aplicaciones subyacentes a cada una de las plataformas, siendo las más conocidas Google Play Store y App Store. Este proceso puede demorar cierta cantidad de tiempo ya que se expone a la app a una revisión por parte de terceros. No es el caso de las aplicaciones web móviles, donde el aplicativo se aloja directamente en un servidor web en la nube sin ser revisado por terceros.

Tecnología de desarrollo seleccionada

Para satisfacer los requerimientos mencionados en la sección anterior se optó por el desarrollo de una PWA. Dicha selección posibilita un desarrollo simple y rápido, donde un mismo código fuente puede ser ejecutado en dispositivos de diferentes características (smartphones, tablets y laptops) mientras éstos cuenten con un navegador web.

Al ser una aplicación web, la puesta en producción y el mantenimiento es relativamente simple. Por otra parte, el aplicativo no está sometido a procesos de revisión de terceros.

En función a una experiencia resiliente u offline, las PWA emplean tecnologías como los Service Workers y almacenamiento caché para poder ejecutarse de manera offline.

Por último las PWA tienen una característica interesante la cual permite a los usuarios instalar en sus dispositivos un acceso directo al aplicativo, simulando una experiencia nativa, como la de una aplicación descargada de una tienda de aplicaciones.

Aspectos técnicos y de implementación

La PWA fue desarrollada utilizando un framework conocido para el desarrollo de aplicaciones multiplataforma híbridas, denominado Ionic Framework [9]. Dicho framework permite desarrollar aplicaciones de software para dispositivos móviles utilizando tecnologías web como el HTML, CSS y JavaScript. En los últimos años Ionic comenzó a dar soporte para crear y desplegar aplicaciones web progresivas utilizando Angular, otro conocido framework que utiliza JavaScript y es de código abierto [10].

El despliegue y puesta a producción se llevaron a cabo utilizando el servicio de hosting de Firebase. El código fuente de la PWA puede encontrarse en [11].

Usuarios del sistema

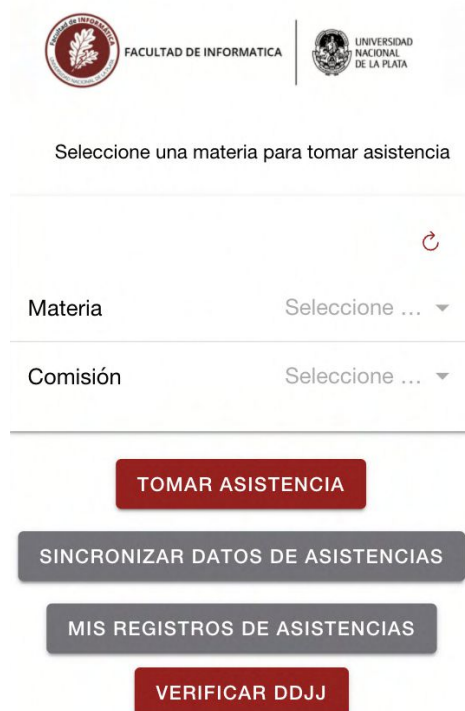
Los usuarios destino del sistema desarrollado fueron los docentes a cargo de cada módulo del curso de ingreso, y el personal no docente a cargo de controlar la entrada al edificio de la Facultad de Informática.

Dichos usuarios, disponían de credenciales de acceso al sistema previamente definidas.

Registro de asistencia al curso de ingreso

La pantalla principal de la PWA desarrollada puede verse en la figura 2. Es el punto de acceso a las dos funcionalidades principales del aplicativo.

En el caso de querer tomar asistencia, los docentes deben seleccionar una materia y comisión. Una vez hecho esto deben presionar el botón “Toma Asistencia”.



Seleccione una materia para tomar asistencia

Materia Seleccione ...

Comisión Seleccione ...

TOMAR ASISTENCIA

SINCRONIZAR DATOS DE ASISTENCIAS

MIS REGISTROS DE ASISTENCIAS

VERIFICAR DDJJ

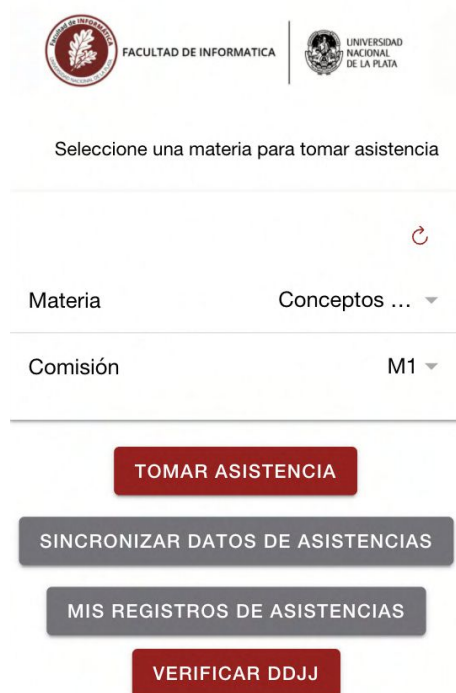


Figura 2. Pantalla principal de la PWA, vista desde un dispositivo móvil tipo smartphone

La pantalla donde se lleva a cabo el proceso la lectura de los códigos QR puede verse en la figura 3. Una vez que la aplicación detecta un código QR, se extrae el DNI, nombre y apellido del estudiante. Dichos datos son almacenados en una estructura de datos en memoria principal mientras no finalice el proceso de la toma de asistencia. Al presionar el botón “Finalizar” las asistencias recolectadas se guardan en el almacenamiento local del dispositivo, pendientes de sincronización con el sistema de gestión de ingreso. Esto permite que el proceso de toma de asistencia se pueda hacer de manera offline.

Al volver a la pantalla principal, figura 4a, el botón “Sincronizar” muestra la cantidad de registros que se encuentran pendientes de comunicar con el sistema de gestión del ingreso. Al presionar dicho

botón, la aplicación sincroniza todos los registros pendientes comunicándose con el sistema de gestión por internet.

Por otra parte, desde la pantalla principal, los usuarios podrán consultar el historial de asistencias registradas con ese dispositivo, presionando sobre el botón “Mis registros de Asistencias”, figura 4b. Accediendo a cada registro se puede visualizar un listado con los datos de los estudiantes registrados y si su asistencia fue sincronizada o no, como se muestra en la figura 4c. Desde esta pantalla se podrá realizar la sincronización de ese registro de asistencia en particular.

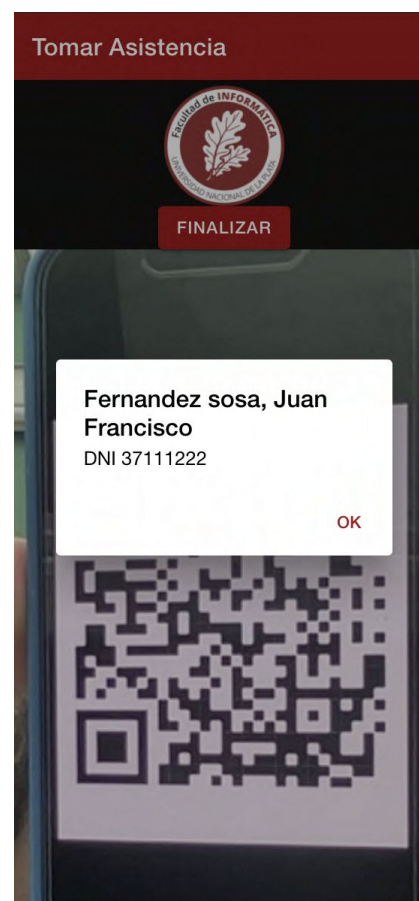


Figura 3. Pantalla de registro de asistencias y detección de códigos QR

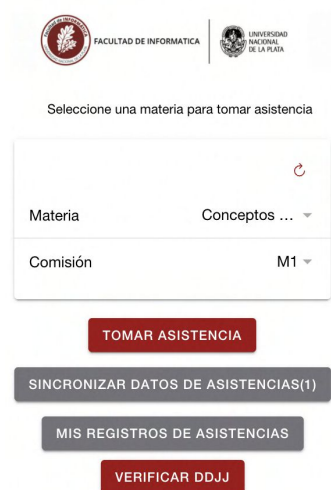


Fig 4a

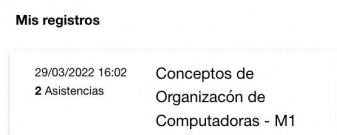


Fig 4b

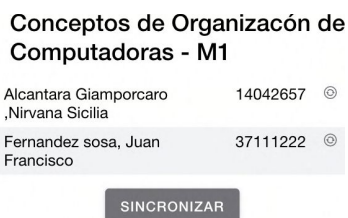


Fig 4c

Figura 4. a) Pantalla principal con registros de asistencia sin sincronizar. b) Listado de registros de asistencia. c) Detalle de un registro de asistencia con el listado de alumnos detectados

Control de declaración jurada de salud

Cómo se mencionó previamente, para volver a las actividades académicas de manera presencial en la UNLP, los

estudiantes deben completar una declaración jurada de salud. En la entrada al edificio, personal no docente debía acreditar esto por medio de la lectura de los mismos códigos QR utilizados para la toma de asistencia.

La declaración jurada de salud fue completada por cada ingresante por medio de un formulario de Google. Este mecanismo permite exportar todas las respuestas del formulario en formato excel o en una hoja de cálculo de Google.

La PWA accede a la hoja de cálculo con los datos de las declaraciones juradas para poder realizar su acreditación. De la hoja de cálculo se extrae un listado de declaraciones juradas realizadas, el cual contiene la fecha de realización y el DNI del estudiante.

Para acreditar si un estudiante realizó la declaración jurada, desde la pantalla principal del aplicativo se debe ingresar con el botón “Verificar DDJJ”, ver figura 2. Esto habilitará el escaneo de los códigos QR. Al detectar un código, se extrae el DNI, y se verifica que el DNI se encuentre dentro del listado de las declaraciones juradas realizadas. De ser así, se muestra en pantalla la fecha de la última declaración jurada realizada (cada estudiante puede realizar más de una), tal como se presenta en la figura 5a. Sino se encuentra el DNI en los registros, no se puede acreditar que se haya realizado una declaración jurada de salud y se informa la situación en pantalla, figura 5b.

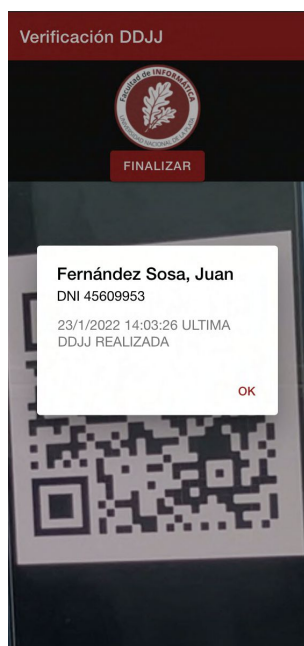


Fig 5a

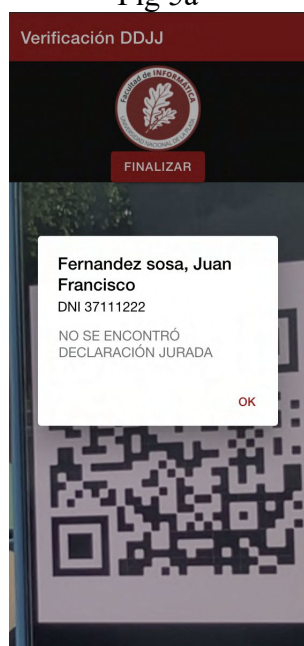


Fig 5b

Figura 5. Pantalla de verificación de declaración jurada de salud. a) Resultado de leer un código QR con un DNI que completó la declaración jurada. b)

Resultado de leer un código QR con un DNI que no completó la declaración jurada

Conclusiones y Trabajo Futuro

Durante la pandemia por COVID-19 las actividades educativas tuvieron que adaptarse al paradigma de dictado de clases

remoto. Luego de atravesar la fase más crítica de la pandemia, se tuvieron que generar protocolos y conductas para la vuelta a la presencialidad en las aulas, en todos los órdenes educativos.

En el curso de ingreso de la Facultad de Informática se solía tomar asistencia por medio del reconocimiento de la huella dactilar de los estudiantes. Esta metodología tuvo que ser reemplazada por una alternativa *contactless*. Particularmente se empleó el uso de códigos QR, tecnología que ya era adoptada en la Facultad para otros fines.

Por otra parte, por una disposición de la Universidad Nacional de La Plata se debía controlar que aquellas personas que ingresen a sus unidades académicas cumplan ciertas condiciones. La Facultad de Informática hizo cumplir estos requerimientos por medio de una declaración jurada de salud.

Para dar soporte al registro de asistencia y a la acreditación de las declaraciones juradas de salud se desarrolló una PWA. La elección de esta tecnología se realizó luego de analizar los requerimientos no funcionales del sistema, entre los cuales se destacan la portabilidad, facilidad de mantenimiento y resiliencia. Este sistema fue utilizado durante el curso de ingreso del 2022 en la Facultad de Informática.

Por último, se logró integrar el sistema de gestión de ingreso que era utilizado en la facultad con este nuevo sistema de manera que, los registros de asistencia se sincronizan automáticamente, acción que antes debía ser realizada manualmente.

Se plantea como trabajo futuro adaptar la aplicación desarrollada de modo que los docentes puedan ingresar con las mismas credenciales que utilizan para el sistema de gestión de ingreso preexistente en la facultad. Esto significa una mayor integración entre los sistemas.

Bibliografía

- [1] Stoyanova-Petrova, S., Kafadarova, N., Stoyanova, D., Mileva, N., Sotirov, S., & Vakrilov, N. (2018). Technology enhanced education by QR codes. In 10th annual International (pp. 6207-6210).
- [2] Fernández Sosa, J., Cuitiño, A., Thomas, P. J., Delía, L. N., Cáseres, G., Corbalán, L. C., & Pesado, P. M. (2017). " Informática UNLP" la App de la Facultad de Informática. In XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017).
- [3] Delía, L., Thomas, P., Corbalan, L., Sosa, J. F., Cuitiño, A., Cáseres, G., & Pesado, P. (2018, Julio). Development approaches for mobile applications: comparative analysis of features. In Science and Information Conference (pp. 470-484). Springer, Cham.
- [4] Thomas, P., Delía, L. N., Corbalan, L., Cáseres, G., Fernández Sosa, J., Tesone, F. & Pesado, P. M. (2018). Tendencias en el desarrollo de Aplicaciones para Dispositivos Móviles. In Workshop de Investigadores en Ciencias de la Computación (Vol. 20).
- [5] Fortunato, D., Bernardino, J.: Progressive web apps: an alternative to the native mobile apps(Portugués). In: 2018 13th Iberian Conference on Information Systems and Technologies(CISTI) (2018)
- [6] Aguirre, V., Delía, L., Thomas, P., Corbalán, L., Cáseres, G., & Sosa, J. F. (2019, October). PWA and TWA: Recent Development Trends. In Argentine Congress of Computer Science (pp. 205-214). Springer, Cham.
- [7] <https://gestioningreso.info.unlp.edu.ar/>
- [8] Vanhala, J. (2017). Implementing an Offline First Web Application
- [9] <https://ionicframework.com/docs>
- [10] <https://angular.io/>
- [11] <https://gitlab.com/jffs/ingreso-qr.git>