

**Departamento de Ciencias de la
Computación (DCCO)**

Carrera de Ingeniería de Software

Prácticas Pre Profesionales (PPP)

Perfil del Proyecto

Presentado por: Cajas Karla

Tutor académico: Ing. Ruíz Jenny

Ciudad: Quito

Fecha: 11/07/2025

ÍNDICE

1. Introducción	5
2. Planteamiento del trabajo.....	5
2.1 Formulación del problema	5
2.2 Justificación.....	6
3. Sistema de Objetivos	6
3.1. Objetivo General.....	6
3.2. Objetivos Específicos (03)	7
4. Alcance	7
5. Marco Teórico	8
5.1 Metodología (Marco de trabajo 5W+2H)	8
6. Ideas a Defender	10
8. Viabilidad (Ej.).....	11
8.1 Humana.....	11
8.1.1 Tutor Empresarial.....	11
8.1.2 Tutor Académico	12
8.1.3 Estudiantes	12
8.2..... Tecnológica	12
8.2.1 Hardware	13
8.2.2 Software.....	13
9. Conclusiones y recomendaciones	13
9.1 Conclusiones	13
9.2 Recomendaciones	13
10. Planificación para el Cronograma	15
11. Referencias.....	15

1. Introducción

En la actualidad, muchas instituciones carecen de sistemas eficientes para validar la identidad de los usuarios mediante métodos biométricos. Esta situación genera procesos lentos, riesgos de suplantación de identidad y una experiencia de usuario poco satisfactoria. Ante este panorama, existe un nicho claro para el desarrollo de soluciones digitales que integren tecnologías de reconocimiento facial, validación de datos en tiempo real y análisis visual de los resultados.

El proyecto Biometrix nace como una solución integral a este problema, combinando tecnologías modernas como Angular, Chart.js, Face-api.js y servicios externos como el registro civil para brindar una experiencia de validación biométrica segura, dinámica y accesible. A partir de un diseño estructurado en Figma, se propone una arquitectura modular con dashboards, formularios inteligentes, reportes, visualización estadística y funcionalidades avanzadas como generación de GIFs, códigos QR personalizados y cambio dinámico de interfaz.

Este enfoque busca cubrir la necesidad de contar con un sistema confiable y extensible que pueda aplicarse en sectores educativos, corporativos o institucionales donde la verificación de identidad sea un requisito esencial.

2. Planteamiento del trabajo

2.1 Formulación del problema

Actualmente, la validación de identidad en muchos entornos institucionales o empresariales depende de métodos manuales o sistemas con tecnologías limitadas, lo que puede generar errores, inseguridad en los datos y una experiencia de usuario deficiente. En el entorno de la empresa donde se desarrolla este proyecto, se ha evidenciado la necesidad de contar con un sistema más preciso y automatizado que permita validar la identidad de las personas de manera rápida, segura y confiable.

Para solucionar estas limitaciones, se ha planteado el desarrollo de Biometrix, una plataforma web que combina tecnologías modernas como Angular y Spring Boot. Esta solución permitirá integrar funcionalidades como reconocimiento facial, formularios validados dinámicamente, generación de gráficos estadísticos en tiempo real y códigos QR personalizados. El sistema también permitirá almacenar la información validada, optimizar el flujo de validación biométrica y garantizar la integridad de los datos de los usuarios.

Además, se incluye la posibilidad de personalizar la interfaz visual del sistema, implementar modales interactivos, almacenar configuraciones localmente y mejorar el rendimiento general de la plataforma, lo cual representa un avance respecto a las soluciones existentes.

El sistema se desarrollará con una arquitectura modular y documentada que facilite su mantenimiento y futura expansión.

2.2 Justificación

Es común encontrar que muchas empresas o instituciones aún validan la identidad de sus usuarios mediante procesos manuales o herramientas básicas, lo que puede ocasionar errores, inseguridad en el manejo de datos y retrasos en la autenticación. Frente a esta realidad, la implementación de un sistema biométrico basado en Angular para el frontend y Spring Boot para el backend se presenta como una solución moderna, eficiente y segura para el control de acceso. El desarrollo de un sistema de este tipo permite automatizar la validación de identidad y centralizar el manejo de la información, minimizando los errores derivados del manejo manual y garantizando la protección de datos sensibles

Esta solución responde a las necesidades de organizaciones que requieren reforzar sus sistemas de seguridad y control de acceso, asegurando una experiencia de usuario rápida, clara y confiable. El uso de tecnologías como Angular permite ofrecer una interfaz dinámica e interactiva, mientras que Spring Boot gestiona de manera segura los procesos de negocio y la protección de la información. La implementación de este sistema biométrico representa una alternativa robusta frente a métodos obsoletos, garantizando precisión, trazabilidad y seguridad en los procesos de autenticación.

3. Sistema de Objetivos

3.1. Objetivo General

Desarrollar una aplicación frontend con Angular que permita automatizar la validación biométrica de identidad mediante una interfaz modular y dinámica, basada en los diseños de Figma. Este sistema facilitará la captura y validación en tiempo real de datos biométricos, integrando reconocimiento facial, formularios validados y visualización estadística, garantizando una experiencia segura, accesible y confiable para los usuarios en entornos institucionales y empresariales.

3.2. Objetivos Específicos (03)

- Implementar una interfaz frontend en Angular, basada en los diseños de Figma, que permita la captura y validación biométrica mediante reconocimiento facial con Face-api.js, generación de GIFs a partir de múltiples fotos y visualización dinámica de estadísticas utilizando Chart.js, ofreciendo una experiencia interactiva y accesible para el usuario.
- Desarrollar un sistema de autenticación y validación de datos biométricos que incluya la integración con servicios externos para la verificación de cédula y código dactilar, garantizando la seguridad, confidencialidad y control de acceso a la información sensible mediante sesiones protegidas y roles de usuario diferenciados.
- Crear módulos de reportes basados en tablas dinámicas que permitan visualizar en tiempo real los registros de validación biométrica organizados según diferentes estados (como éxito, error y en proceso), junto con la generación de códigos QR personalizados.

4. Alcance

El sistema Biometrix permitirá validar la identidad de los usuarios mediante captura y reconocimiento facial, utilizando Angular para el frontend. La aplicación gestionará la captura de imágenes desde la cámara del dispositivo, realizará validaciones faciales con Face-api.js y generará GIFs animados a partir de múltiples fotografías. También permitirá la creación de códigos QR personalizados con la información validada de cada usuario.

En cuanto a la gestión de datos, el sistema mostrará reportes organizados en tablas con diferentes estados (éxito, error, en proceso), facilitando el seguimiento de cada proceso de validación biométrica. La interfaz de usuario se desarrollará respetando fielmente el diseño propuesto por la empresa en Figma, sin modificaciones ni personalizaciones fuera del alcance.

Además, se registrarán logs para el monitoreo de eventos y errores durante el uso de la plataforma. Como complemento, se elaborará documentación técnica detallada que incluya casos de uso y especificaciones funcionales del sistema.

5. Marco Teórico

Visual Studio Code

Visual Studio Code es un editor de código fuente ligero, multiplataforma y altamente personalizable, desarrollado por Microsoft. Es compatible con una amplia variedad de lenguajes de programación, incluyendo TypeScript y JavaScript, que son fundamentales para el desarrollo con Angular. Visual Studio Code facilita la escritura de código mediante funciones como autocompletado inteligente, depuración integrada, control de versiones Git y una gran variedad de extensiones que mejoran la productividad y permiten una navegación eficiente dentro del proyecto.

Angular

Angular es un framework frontend desarrollado por Google para la creación de aplicaciones web dinámicas y de una sola página (SPA). Utiliza TypeScript como lenguaje base y proporciona una arquitectura basada en componentes que facilita la construcción de interfaces de usuario modulares y reutilizables. Angular incorpora herramientas integradas para el manejo de formularios reactivos, gestión de rutas, comunicación con APIs RESTful y la integración de librerías externas, lo que lo convierte en una solución completa para el desarrollo frontend moderno. Su ecosistema permite construir aplicaciones escalables y mantenibles, adaptadas a las necesidades actuales del mercado.

5.1 Metodología (Marco de trabajo 5W+2H)

Debe explicar paso a paso el desarrollo de la guía con la herramienta de Excel aplicando el marco de trabajo de las 5W y 2H

¿QUÉ?	¿CÓMO?	¿QUIÉN?	¿CUÁNDO?	¿POR QUÉ?	¿CUÁNTO?	% DE CUMPLIMIENTO
Introducción del proyecto y documentación de casos de usos V1.0.0	Mediante virtuales	Líder de Desarrollo Josselyn Taco Karla Cajas	07/05/2025 - 08/05/2025	Porque es necesario conocer el contexto del proyecto y documentar sus funcionalidades.	Sin costo	100%
Implementación de validaciones y componentes	Online, desarrollo en Angular	Karla Cajas	08/05/2025 - 09/05/2025	Validar entradas y mostrar	Sin costo	100%

estadísticos con Chart.js	basado en Figma			estadísticas en tiempo real		
Desarrollo de componentes modulares del sistema (dashboard, servicios, formularios)	Online, desde diseño Figma	Karla Cajas	12/05/2025 - 15/05/2025	Modularizar el sistema y permitir una mejor navegación	Sin costo	100%
Implementación de funcionalidades interactivas (modales, formularios dinámicos)	Angular con ReactiveForm sModule	Karla Cajas	16/05/2025	Facilitar experiencia de usuario y capturar información validada	Sin costo	100%
Sistema de cambio de color del sistema (selector, aplicación global, pruebas)	Programación en Angular con estilos dinámicos	Karla Cajas	20/05/2025 - 27/05/2025	Permitir personalización y mejorar la experiencia del usuario	Sin costo	100%
Implementación de validación facial y generación de GIFs	Con Face-api.js, gif.js y cámara	Karla Cajas	28/05/2025 - 03/06/2025	Asegurar autenticidad y enriquecer los datos biométricos	Sin costo	100%
Validación de datos con servicios externos (cédula y código dactilar)	Angular	Karla Cajas	04/06/2025 - 06/06/2025	Verificar identidad real del usuario con fuentes oficiales	Sin costo	100%
Generación de códigos QR personalizados	Uso de qrcode.js	Karla Cajas	09/06/2025 - 10/06/2025	Facilitar identificación rápida y portátil del usuario	Sin costo	100%
Sistema de reportes	Angular + filtros	Karla Cajas	11/06/2025 - 13/06/2025	Brindar seguridad y trazabilidad de las validaciones	Sin costo	100%

Análisis de datos biométricos y retroalimentación visual	Uso de Chart.js / Angular	Karla Cajas	16/06/2025 - 18/06/2025	Obtener métricas e informar al usuario durante la validación	Sin costo	100%
Optimización de reportes y pruebas de carga	Angular + Paginación + Test	Karla Cajas	19/06/2025	Asegurar rendimiento bajo alta demanda	Sin costo	100%
Sistema de logs, gestión de sesiones y pruebas finales	Angular	Karla Cajas Josselyn Taco	20/06/2025 - 25/06/2025	Garantizar trazabilidad y seguridad del sistema	Sin costo	100%

Tabla 1 Marco de trabajo 5W+2H

6. Ideas a Defender

En primer lugar, se defiende el uso del framework Angular como base para construir una interfaz modular, reutilizable y mantenible, la cual permite desarrollar un sistema claro y estructurado de validación biométrica. Angular, al trabajar con componentes independientes y formularios reactivos, facilita la implementación de funcionalidades como la captura de datos, validación facial y visualización de resultados desde una misma interfaz de usuario.

Otra idea clave es la integración de tecnologías modernas para el reconocimiento facial y para la generación de códigos QR personalizados. Estas herramientas permiten construir funcionalidades complejas, reforzando el enfoque de una aplicación ágil, accesible y funcional en tiempo real desde el lado del cliente. Se demuestra que es posible realizar operaciones biométricas directamente desde el navegador, respetando las buenas prácticas de programación y aprovechando al máximo el ecosistema de JavaScript.

Además, se promueve la organización y reutilización del código mediante componentes personalizados, lo cual simplifica el desarrollo de secciones reutilizables y reduce la duplicación de lógica. También se aplica el principio de separación de responsabilidades, manteniendo servicios, componentes y vistas bien delimitados, lo que facilita futuras modificaciones y ampliaciones del sistema.

Finalmente, se destaca la importancia de seguir de forma rigurosa el diseño del interfaz propuesto por la empresa a través de Figma. Esta práctica asegura coherencia

visual, mejora la experiencia del usuario y respeta los lineamientos establecidos por el cliente, cumpliendo con los fundamentos de ingeniería de software en cuanto a diseño centrado en el usuario y documentación visual.

7. Resultados Esperados

- Del proyecto se espera que, a partir del diseño realizado en Figma, se logre aplicar los conocimientos de Fundamentos de Ingeniería de Software y Fundamentos de Programación para desarrollar un sistema funcional. En Ingeniería de Software, se espera fortalecer la habilidad para interpretar requerimientos a partir de un diseño visual y organizar el desarrollo de manera modular y ordenada, aunque sin una arquitectura formal.
- Desde Programación, se busca aplicar principios básicos para construir componentes interactivos y funcionales, gestionar validaciones y estados, y desarrollar una lógica clara y eficiente que permita implementar las funcionalidades planteadas.
- En resumen, se espera obtener un producto que refleje la aplicación práctica de ambos cursos, demostrando la capacidad de transformar un diseño visual en un sistema operativo y mantenible.

8. Viabilidad (Ej.)

Cantidad	Descripción	Valor Unitario (USD)	Valor Total (USD)
	Equipo en casa		
1	Laptop HP 15-ef2xxxxxxx	1300	1300
	Software		
1	Sistema operativo Windows 11 Home	175.50	175.50
1	Visual Studio Code	0	0
1	GitLab	0	0
1	Figma	0	0
		TOTAL	1475.50

Tabla 2 Presupuesto del proyecto

8.1 Humana

8.1.1 Tutor Empresarial

Ing. Bryan Allauca

- **Responsabilidades**

- Líder de Desarrollo en la empresa, encargado de supervisar y coordinar todas las tareas relacionadas con la programación y el desarrollo de software.
- Responsable de la gestión y mantenimiento de proyectos clave como páginas web, el sistema de facturación electrónica y otros desarrollos tecnológicos internos.

8.1.2 Tutor Académico

Ing. Jenny Ruíz.

- **Responsabilidades**

- Docente encargada de guiar al estudiante en la aplicación de la Ingeniería de Software al desarrollo del producto software, supervisando su progreso, promover el desarrollo de habilidades profesionales y evaluar su desempeño durante las prácticas profesionales mediante un informe final.

8.1.3 Estudiantes

Karla Lizbeth Cajas Recalde

- **Responsabilidades**

- Estudiante encargada de cumplir con los tiempos y objetivos establecidos para el desarrollo del proyecto, manteniendo un registro detallado del progreso y avances para entrega del informe final en los horarios establecidos.

8.2 Tecnológica

Para llevar a cabo este proyecto, se ha utilizado un framework frontend moderno que permite construir interfaces dinámicas y modulares. Angular fue elegido por su estructura basada en componentes, lo que facilita la creación de vistas reutilizables, formularios reactivos y la integración con funcionalidades como cámara, validación facial y generación de códigos QR. Esta tecnología permite una comunicación fluida con servicios externos y el manejo eficiente de los datos ingresados por el usuario.

La implementación se desarrolló sobre un diseño previamente establecido en Figma, respetando su estructura visual y jerarquía. La herramienta utilizada para el desarrollo fue Visual Studio Code, por su ligereza, compatibilidad con Angular y facilidad de integración con extensiones útiles para depuración y control de versiones.

Las especificaciones del equipo utilizado se detallan en la sección 8.2.1, y el software empleado durante el desarrollo se describe en la sección 8.2.2.

8.2.1 Hardware

	Requisitos mínimos	Disponibilidad
Memoria RAM	24 GB de RAM	Alta
Almacenamiento	477 GB disco C	Alta

Tabla 3 Requisitos de Hardware

8.2.2 Software

	Requisitos mínimos	Disponibilidad
Sistema Operativo	Se recomienda Windows 11 Home	Alta
IDE	Visual Studio Code es recomendable por su compatibilidad con Angular y funciones como conexión FTP, aunque cualquier otro IDE que cumpla estas características también es válido	Alta

Tabla 4 Requisitos de Software

9. Conclusiones y recomendaciones

9.1 Conclusiones

Se logró desarrollar una aplicación frontend en Angular que cumple con los requisitos de una interfaz modular y dinámica, basada en los diseños de Figma, permitiendo una experiencia de usuario accesible y segura para la validación biométrica de identidad.

- La integración de tecnologías como Face-api.js para reconocimiento facial, Chart.js para visualización de datos y servicios externos para verificación de identidad, permitió implementar funcionalidades completas para la captura, validación y control de acceso a la información, cumpliendo con altos estándares de seguridad y usabilidad.
- Se implementaron módulos de reportes que organizan los registros de validación biométrica según su estado y permiten la generación de códigos QR personalizados, facilitando el monitoreo en tiempo real y mejorando la eficiencia del sistema en entornos institucionales y empresariales.

9.2 Recomendaciones

- Se recomienda continuar desarrollando una integración con un backend seguro para el almacenamiento y gestión de la información validada, así como implementar controles de acceso para proteger los datos sensibles.
- Es importante realizar pruebas exhaustivas en distintos dispositivos y navegadores para garantizar la compatibilidad y correcta funcionalidad del sistema en todos los entornos de uso.

10. Planificación para el Cronograma

#	TAREA	INICIO	FIN
1	Estadísticas y Validaciones	08/05/2025	09/05/2025
2	Componentes modulares desde Figma	12/05/2025	14/05/2025
3	Formularios con validación	15/05/2025	16/05/2025
4	Personalización de color	20/05/2025	27/05/2025
5	Captura y validación facial	04/11/24	25/04/24
6	Generación de GIF facial	02/06/2025	30/05/2025
7	Validación con registro civil	04/06/2025	06/06/2025
8	Código QR con datos del usuario	09/06/2025	10/06/2025
9	Reportes con filtros y ventanas emergentes	11/06/2025	12/06/2025
10	Seguridad, respaldo y autenticación	13/06/2025	16/06/2025
11	Optimización y mejoras finales	17/06/2025	25/06/2025

Tabla 5 Cronograma del proyecto.

11. Referencias

Página oficial de Angular: *Angular. (s/f).* Angular - One framework. Mobile & desktop. Recuperado el 6 de noviembre de 2024, de <https://angular.io/>

Página oficial de Visual Studio Code: *Visual Studio Code. (s/f).* Code editing. Redefined. Recuperado el 6 de noviembre de 2024, de <https://code.visualstudio.com/>