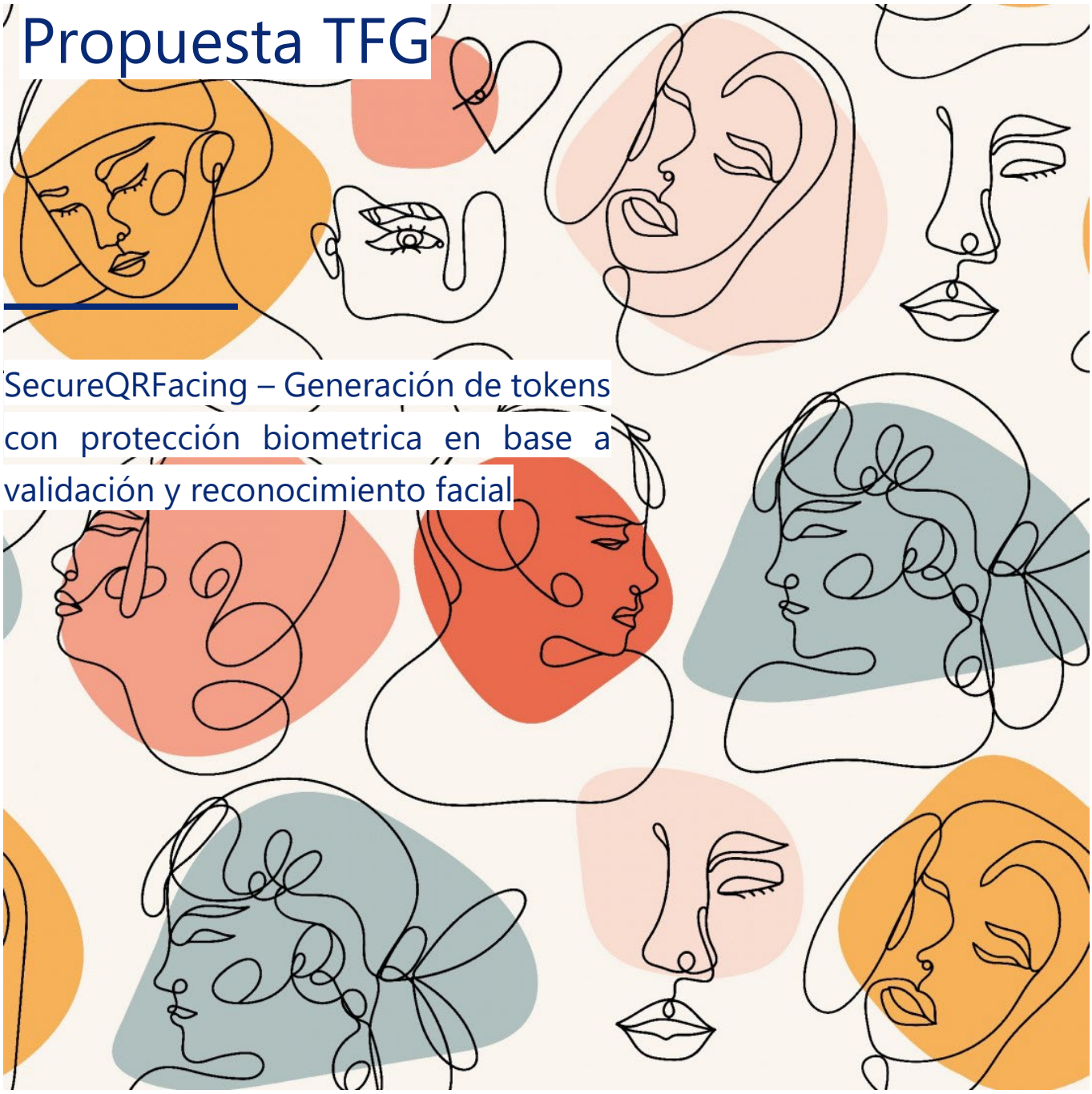
Propuesta TFG



SecureQRFacing – Generación de tokens con protección biometrica en base a validación y reconocimiento facial

6 NOVIEMBRE

Cátedra Masmovil for Advanced Network Engineering and Digital Services

Propuesta de TFG/TFM

Título: SecureQRFacing – Generación de tokens con protección biometrica en base a validación y reconocimiento facial – Parte II

Breve justificación y descripción de la necesidad

En todos los diarios europeos del 29/05/22 se pudieron ver titulares similares al que encontramos en la versión online del periódico VozPópuli, donde se puede leer:

FINAL DE LA CHAMPIONS LEAGUE

Vergüenza de la UEFA en París: caos con las entradas, retrasos y robos a aficionados

Testigos presenciales denuncian robos a aficionados durante el caos con los aficionados del Liverpool en las afueras del Stade de France. Las aglomeraciones retrasaron el partido más de 30 minutos

PUBLICADO 29/05/2022 09:26
ACTUALIZADO 02/06/2022 20:09

"Una vergüenza". La final de la Champions League celebrada anoche en el Stade de France, en París, no sólo pasará a la historia por la hazaña deportiva del Real Madrid, también por **el caos con los miles de hinchas ingleses en los accesos del estadio, los disturbios policiales, las casi 70 detenciones y los más de 170 aficionados que resultaron heridos leves**. El episodio pone bajo lupa a la organización del evento llevada a cabo por la **UEFA**, que no dudó en culpar en un comunicado a los miles de aficionados del Liverpool que acudieron sin entrada. En este ambiente de caos en las afueras del estadio, según relata un testigo presencial, "grupos de franceses de color y magrebíes" aprovecharon para intentar "robar" a los aficionados, incluidos "muchos españoles", que esperaban en los controles.

El alto coste y la impersonalización de las entradas y tickets a los distintos eventos de ocio de cierta relevancia, hace que éstos sean objeto frecuente de manipulación, robo y en general cualquier tipo de fraude.

Esta situación sería fácilmente evitable si los tickets incluyeran algún mecanismo biométrico que permitiera vincular alguna característica física de la persona nominal del ticket, y que dicha caracterización fuera fácilmente validable en el momento de usar el ticket.

Algo parecido ocurre en los billetes de avión, donde éstos incluyen un código de barras o QR que identifica el propio título de vuelo, pero no al individuo o al titular del mismo. El protocolo en el mostrador de embarque exige la validación electrónica del ticket, más una validación adicional (y no vinculada) del pasajero. Esta consiste en la validación de su documento de identificación (DNI, pasaporte) y una comparación ocular de la persona real con el propio documento.

Otros escenarios de uso pueden fácilmente contemplados en el futuro, como por ejemplo la generación de micro-contratos para el turismo vacacional y la identificación de los huéspedes, etc. O incluso la generación de pulseras de ingreso sanitario con firma biométrica: en el ingreso a un servicio de urgencias se genera un código de barras para la pulsera de identificación, pero nada impide que dicha pulsera sea usada por una persona distinta.

La propuesta del proyecto consiste en el estudio, definición e implementación de un mecanismo de firma biométrica que se añada al código QR del ticket en cuestión, y que pueda ser fácilmente validado en tiempo real de su uso. La mejor forma de validación para evitar dispositivos complejos es emplear un *face-id* que permita tanto la captura de la firma biométrica como la validación de la misma mediante otro dispositivo análogo.

El proyecto generará una plataforma denominada "**SecureQRFacing**" (**SQF**) Los mecanismos clave del proyecto puede ser implementados en una plataforma que pueda ser empleada en una nube segura "as-a-service".

Al usuario de un servicio de ticketing que emplee los servicios **SQF** se le podrá ofrecer una opción adicional (con coste) para que haga seguro su ticket. En caso de aceptar, el usuario se tomará una o varias fotos del rostro a través de la aplicación de contratacion (APP en el movil, portal web, o similares). La toma de fotos, puede incluir la toma de una foto del documento de identidad.

Asimismo, la toma de varias fotos permite la realización de una prueba de "fe de vida" que evite el fraude la manipulacion de una foto común para el rostro y el documento de identidad.

El servicio cloud de **SQF** hará:

- **La extracción de los parámetros faciales de las imágenes.**
- **La comparativa entre las imágenes.**
- **La generación de una firma biométrica.**
- **La generación de un código QR con dicha firma.**
- La extracción de los datos del documento de identidad.
- La extracción de la imagen facial del documento de identidad.
- La validación con el documento de identidad, con un cierto rango de precisión que se establezca.
- La generación de un identificador único del usuario y del documento (ofuscado).

El servicio de emisión de tickets incluirá dicho código en el propio ticket.

Posteriomente, cuando el usuario vaya a emplear su ticket (concierto, evento deportivo, avión, etc), el organizador del mismo tomará una foto de la persona portadora del mismo en el momento en que se realiza el escanner del QR y enviará ambos a otro servicio cloud de SQF que procederá a:

- La extracción de la firma biométrica del código QR.
- La extracción biométrica de la foto tomada.
- La comparación entre ambos en base a los umbrales de tolerancia.
- La devolución de un valor de similitud entre ambos.

La propia aplicación del organizador será responsable del posterior tratamiento de dicho dato y del identificador único obtenido.

Existen numerosas variantes al sistema básico que posteriormente serán analizadas y, en algún caso, implementadas, como por ejemplo, la generación de códigos QR biométricos multi-ticket, cuando una persona contrata varios tickets y simplemente una sola los valida a todos (una familia, un grupo de amigos, etc).

Es importante resaltar que este sistema en ningún caso deberá almacenar datos de tipo personal., para evitar problemas legales derivados de la cesión y almacenamiento de datos de carácter temporal.

Objetivos del proyecto

El elemento central del proyecto es una libería de funciones python para el reconocimiento de imagen avanzado, que denominaremos **libSecureQRFacing** que ofrezca los siguientes servicios:

PARTE I (2022-23):

- Servicio de extracción de huellas faciales biométricas.
    - Dada una foto de un sujeto, extraer sus parámetros faciales.
    - Dada una serie de fotos de un supuesto mismo sujeto en diferentes posiciones o ángulos, identificar si se corresponde del mismo sujeto o no, y de si las imágenes se corresponden a la misma ubicación física.
    - Identificacion de posición de imagen facial.
- Servicio de generación de una firma biométrica y su inclusión en un QR.
- Servicio de lectura de QR y extracción de la firma biométrica para su comparación con una imagen o serie de imágenes recién tomadas.

PARTE II (ACTUAL):

- Servicio de escáner de documentos de identidad (OCR):
    - Extracción de datos y validación del documento
    - Reconocimiento de texto.
    - Gestión de posición del documento, ya que normalmente no se encuentra perfectamente alineado y tiene diversos giros.
    - Soporte a multiples tipos de documentos, con algún tipo de sistema de plantillas que simplifique la gestión de:
        - Múltiples formatos de DNIs.
        - Pasaportes
        - Documentos de identidad de otras nacionalidades
    - Soporte a los estándares de documentación electrónica MRZ (ver bibliografía).

Existen numerosos componentes y librería software que realizan uno u otro paso (ver bilbiografía), e incluso hardware específico para la función. Sin embargo, sería muy útil disponer de una librería que unificara los pasos 3 a 8, que denominaremos libcardface lo cual será objetivo del TFG.

Además, como parte fundamental del proyecto, se está trabajando en:

- Optimización de los métodos de reconocimiento facial:

La optimización de los métodos de reconocimiento facial en términos de espacio de representación es crucial para el éxito del proyecto.

Al reducir la dimensionalidad de los datos, se logra una representación más eficiente y manejable de las características faciales (QR's de baja densidad). Esto no solo facilita el procesamiento y almacenamiento de la información, sino que también mejora la velocidad (permite trabajar con **sistemas edge** y equipos menos potentes**)** y precisión del reconocimiento facial.

- Cuantización de la red neuronal.
- Desarrollo de algoritmos de compresión *ad-hoc*.
- Estudio y comparación del desempeño de los mecanismos mencionados.

Dada la complejidad del proyecto, se implementará sólo la librería Python y un pequeño demostrador de funcionamiento.

El proyecto tiene como objetivo el diseño y desarrollo de:

- Una librería PYTHON denominada **libSecureQRFacing** que realice las operaciones mencionadas.
- Un pequeño desarrollo WEB para implementar un demostrador de la tecnología. Se recomienda usar FLASK, UNICORN o similar.
- El demostrador deberá ser desplegado en un contenedor DOCKER autocontenido.

TRABAJOS FUTUROS

Este proyecto será el germen de un proyecto más ambicioso para crear una arquitectura de plataforma, con diversos componentes:

- Una plataforma cloud para SecureQRFacing. Estará formada por:
  - Un conjunto de servicios expresados como API-REST.
  - Un portal de administración de los servicios.
  - Un portal de gestion de usuarios.
- Una aplicación de ejemplo de comercialización.
- Una aplicación de ejemplo de detección y validación.

Tecnologías a emplear

Prototipado en MATLAB

Para realizar prototipos rápidos con los que contrastar los resultados obtenidos se podrá emplear **MATLAB**. y sus librerias de procesamiento avanzado de imagen, así como tecnología de deep-learning para la detección de los patrones documentales.

Librerias de procesamiento de imagen y OCR

En la fase de analisis se realizará un estudio de tecnologías disponibles de procesamiento de imagen para seleccionar la más adecuada para el proyecto. En otras se plantea **TESSERA** o **FACE-RECOGNITION.**

Contenedores DOCKER

Para que el desarrollo tenga un alto grado de portabilidad, se empleará tecnología de contenedores **DOCKER**.

Servidor WEB python

El diseño y su posterior desarrollo será realizado en alguna tecnología de servidor web basada en python, por ejemplo, **FLASK**.

Lenguaje de desarrollo en Python

Todo el desarrollo será implementado en lenguaje **python**, usando alguno de los toolkits de desarrollo más conocido, como puede ser **JetBrains, o VisualStudio**, o similares.

Asimismo se recomienda el uso de herramientas de prototipado rápido como **Anaconda**.

Gestión del ciclo de vida

Todo el ciclo de vida será gestionado mediante servicios **GIT y GITHUB**.

Se emplearán contenedores **Docker y Kubernetes** para la gestión de los entornos y gestión de los despliegues.

Gestión de APIs

Todas las API-REst se expondrán a través de un servicio **SWAGGER**.

Interfaces de usuario

En el caso de tener que implementar interfaces de usuario se empleará el toolkit **REACT**.

Gestión Ágil de Proyectos

Como herramientas de gestión ágil de proyectos se usarán **JIRA y CONFLUENCE**.

Eventualmente se usará SHAREPOINT para toda la documentación del proyecto.

Herramientas de gestión de la productividad

Para toda la parte ofimática se emplearán las herramientas de Office365 suministradas por la UAH.

Gestión de bibliografía

Para una gestión eficiente de la bibliografía, se utilizará **ZOTERO**, una herramienta de gestión de referencias bibliográficas de código abierto. Zotero facilita la organización de referencias y la generación de citas y listas de referencias.

¿A quién va dirigido?

A alumnos de último curso de los grados descritos en los estudios de Ingeniería de Telecomunicación, Ingeniería Telemática, Ingeniería Informática, Ingeniería de Computadores, Ingeniería Industrial o titulaciones similares descritos en:

https://escuelapolitecnica.uah.es/estudios/grados.asp

Alternativamente a alumnos de los másteres universitarios de la Escuela Politécnica Superior descritos en:

https://escuelapolitecnica.uah.es/estudios/masteres-universitarios.asp

¿Qué conocimientos previos se requieren?

Los conocimientos adquiridos durante los estudios requeridos.
Se valorarán especialmente los conocimientos de desarrollo en python.

¿Qué conocimientos y experiencia obtendrá el alumno con el proyecto?

EL proyecto propuesto resuelve una demanda existente en el mercado actualmente, en un escenario completamente real.
La realización del proyecto confiere al alumno unos conocimientos técnicos avanzados que le capacitan laboralmente para la ejecución técnica de proyectos en cualquier compañía presente en el mercado actual, ya que hace hincapié en las tecnologias y métodos más avanzados.
Más allá de las cuestiones meramente técnicas, se dará formación específica al alumno en:

- Metodología y herramientas ágiles.
- Entorno digital de la empresa.

A la terminación del proyecto, el alumno estará plenamente capacitado para su incorporación laboral.

Duración estimada

El proyecto tiene una duracion estimada de 6 meses en dedicacion de media jornada.

Metodología

Todo el proyecto será realizado en metodología agile, empleando herramientas como JIRA y CONFLUENCE, y procurando realizar procesos de integración continua y despliegue continuo (CI/CD).
Se trabajará implementando un mínimo producto viable (MVP) que será refinado en *sprints* cada 2 o 3 semanas.
El alumno dispondrá de un lugar de trabajo específico de la Cátedra en las instalaciones de la UAH, aunque podrá realizar su trabajo de forma semipresencial.
El alumno dispondrá de dos tutores:

- Un tutor académico de la Cátedra en la UAH.
- Un tutor técnico profesional del equipo de MasMovil quien le hará el seguimiento del aprovechamiento de sus trabajos y dará orientación profesional.

Recursos

La Cátedra facilitará un ordenador portátil plataformado y administrado por Grupo MásMóvil para la realización del proyecto. Asimismo, facilitará cuantos recursos y licencias software sean precisos para su ejecución.

Bibliografía y Referencias

https://www.libhunt.com/compare-tesseract-vs-face_recognition
https://github.com/sirfz/tesserocr
https://github.com/ageitgey/face_recognition
https://en.wikipedia.org/wiki/Machine-readable_passport
https://github.com/Arg0s1080/mrz
https://aws.amazon.com/getting-started/cloud-essentials/
https://aws.amazon.com/es/training/
https://www.python.org/
https://www.anaconda.com/products/distribution
https://www.jetbrains.com/es-es/
https://www.jetbrains.com/es-es/pycharm-edu/
https://git-scm.com/
https://github.com/
https://www.atlassian.com/es/software/jira
https://www.atlassian.com/es/software/confluence
https://swagger.io/