# Guió TFG

## Inici

Bon dia, soc el Jaume Casals i avui us presentaré el meu Treball de Final de Grau: Anàlisi automàtica de tiquets de phishing mitjançant el processament del llenguatge natural. Aquest projecte s’ha realitzat dins d’un Conveni de Cooperació Educativa com a part d’un projecte dut a terme per inLab FIB.

## Índex

Parlarem sobre:

1. La contextualització i els objectius del projecte
2. S’explorarà una mica alguns conceptes rellevants
3. La part principal serà el desenvolupament i execució de la pipeline
4. I seguidament l’entrenament dels models d’extracció d’informació
5. Després es mostrarà l’avaluació final dels models entrenats
6. I finalment, es mostrarà el resultat final amb les conclusions del projecte

## Contextualització

### Estadístiques

L'apogeu de la digitalització i la tecnologia ha introduït una comoditat i una connectivitat sense precedents. Però juntament amb aquests avenços, s'ha vist un augment paral·lel de les ciberamenaces i el frau digital. Les estadístiques revelen que les estafes informàtiques denunciades es disparen a més de 300.000 incidents al país només el 2022.

Aquest augment de la ciberdelinqüència no és una anomalia estadística, sinó més aviat un reflex de l'evolució del panorama tecnològic i les seves vulnerabilitats. Les estafes informàtiques han augmentat un 370% en només sis anys, cosa que subratlla la urgència de la nostra resposta a aquestes amenaces.

A mesura que ens enfrontem a aquestes creixents amenaces, es fa imperatiu desenvolupar estratègies proactives per mitigar els riscos i protegir-se contra incidents futurs. Aquest projecte pretén abordar aquesta necessitat analitzant els incidents de phishing i extraient informació clau per millorar la preparació i evitar que es repeteixin.

### Objectiu

L'objectiu principal és dissenyar i desplegar un sistema capaç d’extreure certs camps dels tiquets, cosa que permetrà lluitar de manera proactiva contra les ciberamenaces.

Per aconseguir aquest objectiu general, s'ha marcat els següents objectius:

* Realitzar un estudi de l'estat de l'art investigant les solucions existents per a reptes similars.
* Configurar la base de dades i desenvolupar una eina de descàrrega de tiquets.
* Preprocessar automàticament el text que elimini tota la informació redundant possible i afegeixi al text amb dades rellevants.
* Entrenar un model del processament del llenguatge natural.
* Anonimitzar les dades de sortida, salvaguardant la informació confidencial.
* Configurar i gestionar l'emmagatzematge de les dades preprocessades i anonimitzades en la base de dades de sortida.
* Integrant tots els elements anteriors, implementar una pipeline que processi els tiquets dins de l'entorn especificat.
* Finalment, desenvolupar una API que simplifiqui l'accés i la utilització del sistema.

### Actors implicats

Per explicar resumidament els actors implicats en aquest projecte, l’Agència de ciberseguretat ha creat el projecte global anomenat “TDA”. L’Agència després ha encarregat a i2CAT el seu desenvolupament complet. i2CAT per la seva part, ha confiat en inLab FIB per a dur a terme una part del projecte anomenada “TSC” o “Ticket System Collector”. I aquest és el projecte que s’ha desenvolupat i detallat a la memòria.

### Cronologia

Aquest projecte ha tingut una línia temporal complexa. Primerament, es va iniciar el projecte TSC el desembre del 2022, més d’un any després, es va permetre l'accés a les dades reals amb les quals s'ha de treballar. Addicionalment, es va tardar un mes a obtenir accés a llibreries necessàries per a l'entrenament dels models que s'ha fet servir. Gràcies a l'ampliació del projecte, s'ha pogut redactar i explicar sobre el treball amb els tiquets reals i el desplegament a l’entorn correcte.

## Exploració teòrica

### Anàlisi d’un tiquet

Un tiquet d'incidències és un informe de qualsevol problema o dubte que hagi pogut sorgir, normalment, dins d'una empresa. Aquests tiquets serveixen per comunicar el problema mencionat al tiquet i s'espera obtenir una contestació detallant quins són els passos a seguir per solucionar el problema o una resposta resolent el dubte. Per aquest projecte, s'ha usat OTRS, una eina de gestió i emmagatzematge de tiquets. Gràcies a la reducció de l'abast, només es pot trobar tiquets pertanyents a la categoria de phishing. A continuació, es mostra un tiquet d'exemple d'OTRS amb dos articles amb tots els camps que s’ha de trobar.

En primer lloc, tenim el primer article que escriu una persona consultant sobre un correu sospitós que li ha arribat. Aquí menciona el remitent i l’assumpte del correu i més avall hi ha una adreça electrònica que estava dins del correu. En segon lloc, tenim la resposta on, en aquest cas, la persona que respon ha fet una anàlisi de la situació i contesta amb els usuaris afectats i, al fons, les accions de mitigació (o accions immediates) i accions de control (o accions de prevenció).

### OTRS i Elasticsearch

Com s’ha explicat, l’Agència va proporcionar un servidor amb OTRS, però per fer proves internament abans de rebre accés es va utilitzar Znuny. OTRS tenia una versió gratuïta que va ser discontinuada en un punt i Znuny és un projecte de codi obert que ha continuat el desenvolupament des d’aquell punt.

Per altra banda, l’Agència requereix que s’emmagatzemi la solució en una base de dades Elasticsearch, el qual té un motor de cerca optimitzat per les seves necessitats pròpies. Aquesta base de dades es complementa amb Logstash, una eina de gestió de *logs*, que en aquest cas, permet establir un filtre abans de l’emmagatzematge que anonimitzi els resultats del model.

## Execució pipeline

### Arquitectura

L'arquitectura del sistema està directament lligat a les màquines disponibles de l’Agència i al sistema que tenien en producció. L'Agència va posar a disposició de l'equip tres servidors on estaran les dades d’entrada, les dades de sortida i el servidor on s’executarà la solució. A part, hi ha dos portàtils amb accés als servidors a través d'una VPN privada.

### Pipeline: Extracció de dades

Amb tota aquesta informació, podem començar a explicar la “pipeline” o flux del sistema que s’ha posat en marxa. El primer pas del procés és recuperar els tiquets emmagatzemats dins de la base de dades OTRS. S’accedeix amb PyOTRS el qual permet accedir-hi a través d’una API i ofereix bastanta flexibilitat amb el format del resultat. Aquesta llibertat permet demanar el tiquet en format HTML, el qual és molt convenient perquè es farà servir durant el preprocessament.

### Pipeline: Preprocessament

El preprocessament és tot aquell procés amb l’objectiu de netejar, estandarditzar i preparar les dades per a la ingestió del model. Amb el cos del tiquet en format HTML es poden esbrinar més coses que simplement llegint el text. Per exemple, les respostes dels tiquets que estan repetides, està marcat i es pot eliminar. Per altra banda, també s’analitza el text buscant referències a altres tiquets per veure si es poden afegir al text i aportar informació extra. A més a més, també s’itera sobre els arxius adjunts i es llegeixen aquells més comuns i amb text llegible.

### Preprocessament: Eliminació de signatures

Una part important plantejada del preprocessament és l’eliminació de signatures o peus de pàgina d’un article. Es va notar que en molts correus s’enviava de forma automàtica un missatge amb el telèfon de la persona, un avís sobre la confidencialitat o un recordatori per no imprimir el correu. Aquests finals del correu poden arribar a ocupar fins i tot més que el propi cos de l’article. Per evitar que el missatge principal quedés camuflat i que el model pugui acceptar tiquets més llargs, es va provar diversos mètodes per eliminar-ho.

El primer intent va ser utilitzar models de generació de text com Flan-t5 i FLOR. L’objectiu era donar cada article d’un tiquet individualment al model i que aquest el torni a “generar” sense el text sobrant. En resum, aquests models no van tenir un bon rendiment. Addicionalment, es va comprovar que aquest plantejament tenia els seus problemes com per exemple que el text generat estigui modificat respecte a l’original o que trigui molt temps a fer inferència.

És per això que la següent prova va ser canviant la manera en la qual s’organitzen les dades. En aquest nou dataset, l’article que es vol netejar es divideix en frases fent servir un model del llenguatge natural. Ara l’objectiu és classificar aquestes frases segons si formen part o no de la signatura. Es va provar amb dos models:

* Modificació RoBERTa: (Explicant la imatge) Es crea una representació vectorial de la frase utilitzant el model RoBERTa i fent servir capes denses, es porta tot fins a la capa softmax on escull entre les dues categories. Després de diverses iteracions en l’arquitectura del model, es va aconseguir un 87% de precisió en la classificació de frases.
* Spacy pipeline: en aquest cas es va fer servir un procés ja dissenyat fent servir tok2vec en comptes de transformers com s’ha fet servir fins ara. Tot i això, aquest model va obtenir un 90% de precisió, superant al model RoBERTa.

Aquests resultats, però, van ser insuficients com per considerar-se fiables sense tenir cap pèrdua d’informació i es va decidir no implementar aquesta solució.

### Pipeline: Execució del model

La inferència d’un model ja entrenat normalment és senzilla gràcies a les llibreries que existeixen avui en dia. En aquest cas, quan es rep el text del tiquet, es tokenitza utilitzant el tokenitzador per defecte del model i s’ingereix al model. Aquest genera token per token fins que extreu el token indicador de final de seqüència el que para automàticament la generació i s’obté el JSON resultant.

### Pipeline: Anonimització

Com s’ha mencionat, l’anonimització succeeix a un filtre de Logstash. Aquest mòdul està escoltant a un port fins que arriba el JSON i anonimitza cada camp utilitzant l’algorisme SHA-256 amb l’autenticació HMAC, el que permet afegir una clau que condiciona la generació del hash resultant. Aquest JSON anonimitzat es reparteix en dos índexs de la base de dades Elasticsearch.

### Pipeline: Elasticsearch

La base de dades també està escoltant a un port específic esperant a rebre els resultats anonimitzats i els guarda amb formats específics en índexs concrets. El primer, Events, conté tota la informació anonimitzada d’un tiquet en un mateix document, per altre banda Identities conté documents que contenen una parella clau-valor on es guarda tan la clau hash com el valor original.

### API

També es va crear una API senzilla que permeti l’accés des de fora. Té dos punts d’accés, un que permet processar un individual donant un identificador de tiquet i l’altre que permet processar-ne més seqüencialment llegint els tiquets d’un Excel ja que és així com es van rebre les dades en primera instància.

## Entrenament models

### Dades sintètiques: explicació

A causa de la falta de dades durant el desenvolupament, es va escollir un conjunt de dades que poguessin aproximar-se a les dades reals. Es va trobar un conjunt de dades que consisteix en el Reconeixement d'Entitats a Sentències Judicials de l'Índia. S’esperava que fossin dades similars perquè són entitats que depenen del seu context i d'una llargada similar. Però en comptes de resoldre aquesta tasca com un reconeixement d'entitats, s'ha fet servir generació de text igual que es faria servir amb la solució final.

### Dades sintètiques: Models destacats

Aquests són els models que majoritàriament es van fer servir durant el desenvolupament. Vam utilitzar l’eina LocalGPT per provar de manera local els models de Mistral 7B i Llama 2 els quals eren molt avançats quan van sortir. El problema d’aquests és que tenen poc coneixement multilingüe i no teníem suficient memòria en la GPU per entrenar tots els paràmetres d’aquests models.

Per altra banda, es va trobar la família Flan-T5. Aquest model desenvolupat per Google està basat en un sistema encoder-decoder amb una mida relativament petita. Va ser entrenat per complir més de 1.800 tasques i per entendre més de 50 idiomes. Es va entrenar tant el Flan-T5 base com una versió fine-tunejada amb el conjunt de dades “LaMini”. Es va provar també les diferents mides i, en conclusió la versió que va aconseguir uns encerts més elevats va ser el model Flan-T5-base “LaMini”.

### Dades reals: explicació

Les dades reals consisteixen en els tiquets de phishing que han sigut explicats abans amb els seus 7 camps a trobar. Quan un dels camps no es pot trobar textualment dins del contingut del tiquet, s'escriu "NotFound" en comptes de la resposta. Analitzant els 95 tiquets que es van etiquetar a mà, es va trobar que entre un 60 i 70% dels camps són "NotFound". Aquest fet pot tenir implicacions negatives per al model, perquè això significa que la majoria dels tiquets que veu durant l'entrenament no li serveixen per entendre que significa cada camp i, per tant, necessita moltes més dades per aconseguir un resultat satisfactori.

### Dades reals: Models destacats

Aquestes dades es van observar més complexes, més llargues de mitjana i, el que va reduir més l'espai de cerca, és que la gran majoria de tiquets estaven en català. És per aquests motius que els models fets servir anteriorment com el Mistral 7b, el Llama 2 i el flan-t5 ja no eren prou capaços per extreure els camps dels tiquets. Llavors, es va tornar a investigar models i els que més van prometre van ser els models FLOR i el QWEN1.5. El primer és un reentrenament del model BLOOM 6.3B fet pel projecte AINA on el conjunt de dades conté en parts iguals català, castellà i anglès. Per altra banda, Qwen1.5 és la segona versió del model desenvolupat pel grup Alibaba que principalment està entrenat en anglès i xinès, però també té molt bona capacitat multilingüe.

## Avaluació dels models

### Resultats dels models

El model FLOR de sis mil tres-cents milions de paràmetres no va ser capaç de generar una sortida en format JSON. Es creu que aquest model està més enfocat a la generació de textos i no entén el format de resposta.

El model QWEN1.5 amb set mil milions de paràmetres, en canvi, pot generar de manera confiable els resultats amb el format esperat. No només això, sinó que també aconsegueix un percentatge d'encerts elevat a excepció d'alguns camps on voreja el 50% d'encerts. En general, té una precisió de més del 70% el qual es considera suficient tenint en compte els limitats recursos amb els quals s'ha disposat.

### Percentatge de “NotFound”

L'altra cara de la moneda és quan s'analitza quins resultats ha tret el model. Com es pot veure, la majoria dels camps no han sigut trobats i això simplement reflecteix les dades amb les quals ha sigut entrenat, ja que, al final, el model només és el seu conjunt d'entrenament representat en els seus pesos. Per exemple les accions de control coincideix que hi ha pràcticament un 100% de no trobats tant en el conjunt de dades com les prediccions amb el conjunt de test. Però altres com el Mail de la víctima que mostra que encara hi ha espai de millora.

## Conclusions

Al final, durant aquest projecte s'ha implementat la pipeline de l'extracció d'informació. Per culpa de no tenir GPU, en el moment d'escriure la memòria la implementació va ser la que es veu en pantalla. La pipeline s'inicia des del portàtil amb GPU des d'on recull un tiquet la base de dades OTRS. En el mateix portàtil es preprocessa i s'executa el model i s'envia mitjançant SSH el JSON resultant a la màquina que conté la base de dades Elasticsearch. Aquesta màquina s'encarrega d'anonimitzar i repartir els documents resultants als índexs correctes.