



# Instituto Superior de Tecnologias Avançadas

Licenciatura em Engenharia Informática

PROJECTO GLOBAL

CRYPTOPROXY

**Aluno:** Tiago Alexandre D. M. Correia  
AL 1733 – Turma A – Lisboa  
Ano Letivo 2012/13

**Coordenador:** Prof. Dr. Pedro Brandão

**Orientador:** Prof. Dr. Pedro Brandão

Trabalho de conclusão de curso  
de Engenharia de Informática  
na área de Engenharia de Software

2014

ROBERTO SILVA

2014

Trabalho de conclusão de curso  
de Engenharia de Informática  
na área de Engenharia de Software  
na Universidade Tecnológica Federal do Paraná  
em Curitiba, Paraná, Brasil

**Trabalho redigido com o acordo ortográfico**



## AGRADECIMENTOS

*Dedico este trabalho aos meus Pais, por todo o apoio que me deram no início da minha vida, ao “alimentarem” constantemente a minha vontade em aprender com as tecnologias da informação e pelo empenho e sacrifício que decidiram ter para eu ter tudo ao meu alcance.*

... (mirrored text from reverse side) ...

... (mirrored text from reverse side) ...

- 1. ...
- 2. ...
- 3. ...

... (mirrored text from reverse side) ...

... (mirrored text from reverse side) ...

## RESUMO

Neste projeto, realizado no âmbito do trabalho final de curso, iniciado e finalizado no ano letivo de 2017/2018, do curso de Engenharia Informática do Instituto Superior de Tecnologias Avançadas (ISTEC), tem como objectivo a demonstração do desenvolvimento de uma aplicação de transmissão de dados, que recorre a um método criptográfico para a proteção dos dados e utilizando o sistema operativo Windows 8.1/10 Enterprise Edition.

A aplicação tem como foco principal a operacionalidade de um *Cryptoproxy*<sup>1</sup>, cujas funções principais enumeram-se da seguinte forma:

1. Implementação da segurança da informação (Encriptação e Desencriptação de dados);
2. Confidencialidade, integridade, disponibilidade e segurança na transmissão remota de dados;
3. Manuseamento e tratamento de dados em matrizes e visualização gráfica.

É importante acrescentar que este projeto foi elaborado de acordo com a atual necessidade em se implementar nos serviços informáticos e administrativos de uma instituição bancária, uma aplicação departamental para controlo de informação de alto nível de sensibilidade.

Palavras-chave: *Windows 8.1/10, Segurança, Encriptação, Desencriptação, Confidencialidade, integridade, disponibilidade*

<sup>1</sup> Fornecedor de interface de segurança para funcionalidades criptográficas.



In this process, students are required to work in groups and to prepare a presentation on the topic of the lesson. The teacher should provide support and guidance to the students during this process. The teacher should also provide feedback to the students on their presentations.

The teacher should also provide feedback to the students on their presentations.

1. The teacher should provide feedback to the students on their presentations.
2. The teacher should provide feedback to the students on their presentations.
3. The teacher should provide feedback to the students on their presentations.

It is important to note that the teacher should provide feedback to the students on their presentations. The teacher should provide feedback to the students on their presentations.

The teacher should also provide feedback to the students on their presentations.

## ABSTRACT

In this project, undertaken as part of the final course, started and completed in the academic year of 2016/2017, the Bachelor's Degree in Computer Engineering from the Advanced Technology Institute (ISTEC), it has a goal to demonstrate the development of an data transmission application with data encryption for data protection and using the Windows 8.1/10 Enterprise Edition operating system.

The application has as main focus the operability of a *Cryptoproxy*<sup>1</sup>, which main functions are enumerated as follows:

1. Implementation of information security (data encrypting e decrypting);
2. Speed on the encryption and decryption of information;
3. Automation in the selection and analyses of appropriate encryption methods to information.

It is important to add that this project was elaborated with the actual need to implement, in an information e administrative branch of a bank institution, a specific branch application to control information of a high sensitivity level.

Keywords: Windows 10, Security, Encryption, Decryption, Information Security, *Confidentiality, integrity, availability*

<sup>1</sup> Secure interface provider for the cryptographic functionality

CONTENTS

1. Introduction	1
2. Objectives of the Project	2
3. Methodology	3
4. Results and Discussion	4
5. Conclusion	5
6. References	6
7. Appendix	7
8. Bibliography	8
9. Glossary	9
10. Acknowledgements	10
11. Declaration	11
12. Certificate	12
13. Appendix A	13
14. Appendix B	14
15. Appendix C	15
16. Appendix D	16
17. Appendix E	17
18. Appendix F	18
19. Appendix G	19
20. Appendix H	20
21. Appendix I	21
22. Appendix J	22
23. Appendix K	23
24. Appendix L	24
25. Appendix M	25
26. Appendix N	26
27. Appendix O	27
28. Appendix P	28
29. Appendix Q	29
30. Appendix R	30
31. Appendix S	31
32. Appendix T	32
33. Appendix U	33
34. Appendix V	34
35. Appendix W	35
36. Appendix X	36
37. Appendix Y	37
38. Appendix Z	38

## Abreviaturas e notações adotadas

CIA – Central Intelligence Agency

NSA – National Security Agency

S.O. – Sistema Operativo

AES – Advanced Encryption Standard

DES – Data Encryption Standard

RSA – Rivest Shamir e Adleman

CSP – Cryptographic Services Providers

C – linguagem de programação

C++ - C plusplus, linguagem de programação

C# - C Sharp, linguagem de programação

HTML - Hypertext Markup Language

JavaScript – Scripting Language

OOP – Object Oriented Programming, programação orientada por objetos

CSS – Cascading Style Sheets

BASIC - Beginner's All-purpose Symbolic Instruction Code, linguagem de programação

Scripting – Linguagem de programação de automatização de rotinas

TCP – Transport Control Protocol, protocolo de controlo e transporte de dados informáticos

IP – Internet Protocol, protocolo de internet que define o endereçamento nas redes

TPA – Terminal de Pagamento Automático

ATM – Automatic Transfer Machine, máquinas de pagamento automático

MIT – Massachusetts Institute of Technology

PT – Posto de Trabalho

1	INTRODUCTION
2	1.1 - OBJETIVO
3	1.2 - ALCANCE
4	1.3 - DEFINICIONES
5	1.4 - REFERENCIAS
6	2. - OBJETIVO
7	3. - ALCANCE
8	4. - DEFINICIONES
9	5. - REFERENCIAS
10	6. - OBJETIVO
11	7. - ALCANCE
12	8. - DEFINICIONES
13	9. - REFERENCIAS
14	10. - OBJETIVO
15	11. - ALCANCE
16	12. - DEFINICIONES
17	13. - REFERENCIAS
18	14. - OBJETIVO
19	15. - ALCANCE
20	16. - DEFINICIONES
21	17. - REFERENCIAS
22	18. - OBJETIVO
23	19. - ALCANCE
24	20. - DEFINICIONES
25	21. - REFERENCIAS
26	22. - OBJETIVO
27	23. - ALCANCE
28	24. - DEFINICIONES
29	25. - REFERENCIAS
30	26. - OBJETIVO
31	27. - ALCANCE
32	28. - DEFINICIONES
33	29. - REFERENCIAS
34	30. - OBJETIVO
35	31. - ALCANCE
36	32. - DEFINICIONES
37	33. - REFERENCIAS
38	34. - OBJETIVO
39	35. - ALCANCE
40	36. - DEFINICIONES
41	37. - REFERENCIAS
42	38. - OBJETIVO
43	39. - ALCANCE
44	40. - DEFINICIONES
45	41. - REFERENCIAS
46	42. - OBJETIVO
47	43. - ALCANCE
48	44. - DEFINICIONES
49	45. - REFERENCIAS
50	46. - OBJETIVO
51	47. - ALCANCE
52	48. - DEFINICIONES
53	49. - REFERENCIAS
54	50. - OBJETIVO
55	51. - ALCANCE
56	52. - DEFINICIONES
57	53. - REFERENCIAS
58	54. - OBJETIVO
59	55. - ALCANCE
60	56. - DEFINICIONES
61	57. - REFERENCIAS
62	58. - OBJETIVO
63	59. - ALCANCE
64	60. - DEFINICIONES
65	61. - REFERENCIAS
66	62. - OBJETIVO
67	63. - ALCANCE
68	64. - DEFINICIONES
69	65. - REFERENCIAS
70	66. - OBJETIVO
71	67. - ALCANCE
72	68. - DEFINICIONES
73	69. - REFERENCIAS
74	70. - OBJETIVO
75	71. - ALCANCE
76	72. - DEFINICIONES
77	73. - REFERENCIAS
78	74. - OBJETIVO
79	75. - ALCANCE
80	76. - DEFINICIONES
81	77. - REFERENCIAS
82	78. - OBJETIVO
83	79. - ALCANCE
84	80. - DEFINICIONES
85	81. - REFERENCIAS
86	82. - OBJETIVO
87	83. - ALCANCE
88	84. - DEFINICIONES
89	85. - REFERENCIAS
90	86. - OBJETIVO
91	87. - ALCANCE
92	88. - DEFINICIONES
93	89. - REFERENCIAS
94	90. - OBJETIVO
95	91. - ALCANCE
96	92. - DEFINICIONES
97	93. - REFERENCIAS
98	94. - OBJETIVO
99	95. - ALCANCE
100	96. - DEFINICIONES
101	97. - REFERENCIAS
102	98. - OBJETIVO
103	99. - ALCANCE
104	100. - DEFINICIONES
105	101. - REFERENCIAS
106	102. - OBJETIVO
107	103. - ALCANCE
108	104. - DEFINICIONES
109	105. - REFERENCIAS
110	106. - OBJETIVO
111	107. - ALCANCE
112	108. - DEFINICIONES
113	109. - REFERENCIAS
114	110. - OBJETIVO
115	111. - ALCANCE
116	112. - DEFINICIONES
117	113. - REFERENCIAS
118	114. - OBJETIVO
119	115. - ALCANCE
120	116. - DEFINICIONES
121	117. - REFERENCIAS
122	118. - OBJETIVO
123	119. - ALCANCE
124	120. - DEFINICIONES
125	121. - REFERENCIAS
126	122. - OBJETIVO
127	123. - ALCANCE
128	124. - DEFINICIONES
129	125. - REFERENCIAS
130	126. - OBJETIVO
131	127. - ALCANCE
132	128. - DEFINICIONES
133	129. - REFERENCIAS
134	130. - OBJETIVO
135	131. - ALCANCE
136	132. - DEFINICIONES
137	133. - REFERENCIAS
138	134. - OBJETIVO
139	135. - ALCANCE
140	136. - DEFINICIONES
141	137. - REFERENCIAS
142	138. - OBJETIVO
143	139. - ALCANCE
144	140. - DEFINICIONES
145	141. - REFERENCIAS
146	142. - OBJETIVO
147	143. - ALCANCE
148	144. - DEFINICIONES
149	145. - REFERENCIAS
150	146. - OBJETIVO
151	147. - ALCANCE
152	148. - DEFINICIONES
153	149. - REFERENCIAS
154	150. - OBJETIVO
155	151. - ALCANCE
156	152. - DEFINICIONES
157	153. - REFERENCIAS
158	154. - OBJETIVO
159	155. - ALCANCE
160	156. - DEFINICIONES
161	157. - REFERENCIAS
162	158. - OBJETIVO
163	159. - ALCANCE
164	160. - DEFINICIONES
165	161. - REFERENCIAS
166	162. - OBJETIVO
167	163. - ALCANCE
168	164. - DEFINICIONES
169	165. - REFERENCIAS
170	166. - OBJETIVO
171	167. - ALCANCE
172	168. - DEFINICIONES
173	169. - REFERENCIAS
174	170. - OBJETIVO
175	171. - ALCANCE
176	172. - DEFINICIONES
177	173. - REFERENCIAS
178	174. - OBJETIVO
179	175. - ALCANCE
180	176. - DEFINICIONES
181	177. - REFERENCIAS
182	178. - OBJETIVO
183	179. - ALCANCE
184	180. - DEFINICIONES
185	181. - REFERENCIAS
186	182. - OBJETIVO
187	183. - ALCANCE
188	184. - DEFINICIONES
189	185. - REFERENCIAS
190	186. - OBJETIVO
191	187. - ALCANCE
192	188. - DEFINICIONES
193	189. - REFERENCIAS
194	190. - OBJETIVO
195	191. - ALCANCE
196	192. - DEFINICIONES
197	193. - REFERENCIAS
198	194. - OBJETIVO
199	195. - ALCANCE
200	196. - DEFINICIONES
201	197. - REFERENCIAS
202	198. - OBJETIVO
203	199. - ALCANCE
204	200. - DEFINICIONES
205	201. - REFERENCIAS
206	202. - OBJETIVO
207	203. - ALCANCE
208	204. - DEFINICIONES
209	205. - REFERENCIAS
210	206. - OBJETIVO
211	207. - ALCANCE
212	208. - DEFINICIONES
213	209. - REFERENCIAS
214	210. - OBJETIVO
215	211. - ALCANCE
216	212. - DEFINICIONES
217	213. - REFERENCIAS
218	214. - OBJETIVO
219	215. - ALCANCE
220	216. - DEFINICIONES
221	217. - REFERENCIAS
222	218. - OBJETIVO
223	219. - ALCANCE
224	220. - DEFINICIONES
225	221. - REFERENCIAS
226	222. - OBJETIVO
227	223. - ALCANCE
228	224. - DEFINICIONES
229	225. - REFERENCIAS
230	226. - OBJETIVO
231	227. - ALCANCE
232	228. - DEFINICIONES
233	229. - REFERENCIAS
234	230. - OBJETIVO
235	231. - ALCANCE
236	232. - DEFINICIONES
237	233. - REFERENCIAS
238	234. - OBJETIVO
239	235. - ALCANCE
240	236. - DEFINICIONES
241	237. - REFERENCIAS
242	238. - OBJETIVO
243	239. - ALCANCE
244	240. - DEFINICIONES
245	241. - REFERENCIAS
246	242. - OBJETIVO
247	243. - ALCANCE
248	244. - DEFINICIONES
249	245. - REFERENCIAS
250	246. - OBJETIVO
251	247. - ALCANCE
252	248. - DEFINICIONES
253	249. - REFERENCIAS
254	250. - OBJETIVO
255	251. - ALCANCE
256	252. - DEFINICIONES
257	253. - REFERENCIAS
258	254. - OBJETIVO
259	255. - ALCANCE
260	256. - DEFINICIONES
261	257. - REFERENCIAS
262	258. - OBJETIVO
263	259. - ALCANCE
264	260. - DEFINICIONES
265	261. - REFERENCIAS
266	262. - OBJETIVO
267	263. - ALCANCE
268	264. - DEFINICIONES
269	265. - REFERENCIAS
270	266. - OBJETIVO
271	267. - ALCANCE
272	268. - DEFINICIONES
273	269. - REFERENCIAS
274	270. - OBJETIVO
275	271. - ALCANCE
276	272. - DEFINICIONES
277	273. - REFERENCIAS
278	274. - OBJETIVO
279	275. - ALCANCE
280	276. - DEFINICIONES
281	277. - REFERENCIAS
282	278. - OBJETIVO
283	279. - ALCANCE
284	280. - DEFINICIONES
285	281. - REFERENCIAS
286	282. - OBJETIVO
287	283. - ALCANCE
288	284. - DEFINICIONES
289	285. - REFERENCIAS
290	286. - OBJETIVO
291	287. - ALCANCE
292	288. - DEFINICIONES
293	289. - REFERENCIAS
294	290. - OBJETIVO
295	291. - ALCANCE
296	292. - DEFINICIONES
297	293. - REFERENCIAS
298	294. - OBJETIVO
299	295. - ALCANCE
300	296. - DEFINICIONES
301	297. - REFERENCIAS
302	298. - OBJETIVO
303	299. - ALCANCE
304	300. - DEFINICIONES
305	301. - REFERENCIAS
306	302. - OBJETIVO
307	303. - ALCANCE
308	304. - DEFINICIONES
309	305. - REFERENCIAS
310	306. - OBJETIVO
311	307. - ALCANCE
312	308. - DEFINICIONES
313	309. - REFERENCIAS
314	310. - OBJETIVO
315	311. - ALCANCE
316	312. - DEFINICIONES
317	313. - REFERENCIAS
318	314. - OBJETIVO
319	315. - ALCANCE
320	316. - DEFINICIONES
321	317. - REFERENCIAS
322	318. - OBJETIVO
323	319. - ALCANCE
324	320. - DEFINICIONES
325	321. - REFERENCIAS
326	322. - OBJETIVO
327	323. - ALCANCE
328	324. - DEFINICIONES
329	325. - REFERENCIAS
330	326. - OBJETIVO
331	327. - ALCANCE
332	328. - DEFINICIONES
333	329. - REFERENCIAS
334	330. - OBJETIVO
335	331. - ALCANCE
336	332. - DEFINICIONES
337	333. - REFERENCIAS
338	334. - OBJETIVO
339	335. - ALCANCE
340	336. - DEFINICIONES
341	337. - REFERENCIAS
342	338. - OBJETIVO
343	339. - ALCANCE
344	340. - DEFINICIONES
345	341. - REFERENCIAS
346	342. - OBJETIVO
347	343. - ALCANCE
348	344. - DEFINICIONES
349	345. - REFERENCIAS
350	346. - OBJETIVO
351	347. - ALCANCE
352	348. - DEFINICIONES
353	349. - REFERENCIAS
354	350. - OBJETIVO
355	351. - ALCANCE
356	352. - DEFINICIONES
357	353. - REFERENCIAS
358	354. - OBJETIVO
359	355. - ALCANCE
360	356. - DEFINICIONES
361	357. - REFERENCIAS
362	358. - OBJETIVO
363	359. - ALCANCE
364	360. - DEFINICIONES
365	361. - REFERENCIAS
366	362. - OBJETIVO
367	363. - ALCANCE
368	364. - DEFINICIONES
369	365. - REFERENCIAS
370	366. - OBJETIVO
371	367. - ALCANCE
372	368. - DEFINICIONES
373	369. - REFERENCIAS
374	370. - OBJETIVO
375	371. - ALCANCE
376	372. - DEFINICIONES
377	373. - REFERENCIAS
378	374. - OBJETIVO
379	375. - ALCANCE
380	376. - DEFINICIONES
381	377. - REFERENCIAS
382	378. - OBJETIVO
383	379. - ALCANCE
384	380. - DEFINICIONES
385	381. - REFERENCIAS
386	382. - OBJETIVO
387	383. - ALCANCE
388	384. - DEFINICIONES
389	385. - REFERENCIAS
390	386. - OBJETIVO
391	387. - ALCANCE
392	388. - DEFINICIONES
393	389. - REFERENCIAS
394	390. - OBJETIVO
395	391. - ALCANCE
396	392. - DEFINICIONES
397	393. - REFERENCIAS
398	394. - OBJETIVO
399	395. - ALCANCE
400	396. - DEFINICIONES
401	397. - REFERENCIAS
402	398. - OBJETIVO
403	399. - ALCANCE
404	400. - DEFINICIONES
405	401. - REFERENCIAS
406	402. - OBJETIVO
407	403. - ALCANCE
408	404. - DEFINICIONES
409	405. - REFERENCIAS
410	406. - OBJETIVO
411	407. - ALCANCE
412	408. - DEFINICIONES
413	409. - REFERENCIAS
414	410. - OBJETIVO
415	411. - ALCANCE
416	412. - DEFINICIONES
417	413. - REFERENCIAS
418	414. - OBJETIVO
419	415. - ALCANCE
420	416. - DEFINICIONES
421	417. - REFERENCIAS
422	418. - OBJETIVO
423	419. - ALCANCE
424	420. - DEFINICIONES
425	421. - REFERENCIAS
426	422. - OBJETIVO
427	423. - ALCANCE
428	424. - DEFINICIONES
429	425. - REFERENCIAS
430	426. - OBJETIVO
431	427. - ALCANCE
432	428. - DEFINICIONES
433	429. - REFERENCIAS
434	430. - OBJETIVO
435	431. - ALCANCE
436	432. - DEFINICIONES
437	433. - REFERENCIAS
438	434. - OBJETIVO
439	435. - ALCANCE
440	436. - DEFINICIONES
441	437. - REFERENCIAS
442	438. - OBJETIVO
443	439. - ALCANCE
444	440. - DEFINICIONES
445	441. - REFERENCIAS
446	442. - OBJETIVO
447	443. - ALCANCE
448	444. - DEFINICIONES
449	445. - REFERENCIAS
450	446. - OBJETIVO
451	447. - ALCANCE
452	448. - DEFINICIONES
453	449. - REFERENCIAS
454	450. - OBJETIVO
455	451. - ALCANCE
456	452. - DEFINICIONES
457	453. - REFERENCIAS
458	454. - OBJETIVO
459	455. - ALCANCE
460	456. - DEFINICIONES
461	457. - REFERENCIAS
462	458. - OBJETIVO
463	459. - ALCANCE
464	460. - DEFINICIONES
465	461. - REFERENCIAS
466	462. - OBJETIVO
467	463. - ALCANCE
468	464. - DEFINICIONES
469	465. - REFERENCIAS
470	466. - OBJETIVO
471	467. - ALCANCE
472	468. - DEFINICIONES
473	469. - REFERENCIAS
474	470. - OBJETIVO
475	471. - ALCANCE
476	472. - DEFINICIONES
477	473. - REFERENCIAS
478	474. - OBJETIVO
479	475. - ALCANCE
480	476. - DEFINICIONES
481	477. - REFERENCIAS
482	478. - OBJETIVO
483	479. - ALCANCE
484	480. - DEFINICIONES
485	481. - REFERENCIAS
486	482. - OBJETIVO
487	483. - ALCANCE
488	484. - DEFINICIONES
489	485. - REFERENCIAS
490	486. - OBJETIVO
491	487. - ALCANCE
492	488. - DEFINICIONES
493	489. - REFERENCIAS
494	490. - OBJETIVO
495	491. - ALCANCE
496	492. - DEFINICIONES
497	493. - REFERENCIAS
498	494. - OBJETIVO
499	495. - ALCANCE
500	496. - DEFINICIONES

## Índice

AGRADECIMENTOS .....	iii
RESUMO .....	v
ABSTRACT .....	vii
Abreviaturas e notações adotadas .....	ix
Índice.....	xi
Índice de figuras .....	xii
INTRODUÇÃO .....	1
Enquadramento .....	4
Organização do Projeto Global.....	6
2 - ESTADO DA ARTE.....	8
2.1 – Introdução à história da criptografia .....	8
2.2 - Criptografia Clássica .....	8
2.3 - Criptografia Moderna.....	12
2.3.1 - Resumo .....	15
2.4 - Cryptoproxy.....	16
3 – Linguagens de programação.....	17
3.1 - Resumo .....	28
4 – Protocolo de Comunicação.....	29
3 - Apresentação da aplicação – Cryptoproxy (PT cliente e PT servidor).....	31
3.1 – Características da Aplicação .....	31
3.2 – Funcionalidades da Aplicação.....	32
Conclusão.....	38
Webgrafia.....	39

## Índice de figuras

Figura 1 - scytale (bastão) ou cícala espartana .....	9
Figura 2 - cifra de César (Júlio César).....	9
Figura 3 - cifra de Vigenére .....	10
Figura 4 - cifra Playfair.....	10
Figura 5 - máquina Enigma.....	11
Figura 6 - cifra DES (esquema funcional).....	12
Figura 7 - cifra AES (esquema funcional).....	13
Figura 8 - cifra RSA (esquema funcional).....	14
Figura 9 - cartão com chip (smart card).....	16
Figura 10 - Top Ten Languages 2017 .....	28
Figura 11 - versão "PT cliente" .....	32
Figura 12 - seleccionar ficheiro de texto .....	32
Figura 13 - introduzir IP "PT servidor" .....	33
Figura 14 - envio e tempo de transmissão .....	33
Figura 15 - versão "PT servidor" .....	34
Figura 16 - seleccionar pasta de destino .....	34
Figura 17 - ficheiro recebido encriptado .....	35
Figura 18 - ficheiro desencriptado.....	35
Figura 19 - visualizar informação recebida .....	36
Figura 20 - análise AAC estatística.....	36
Figura 21 - análise AAC gráfica .....	37



## INTRODUÇÃO

Lembro-me que em 2013, mais precisamente a meio do semestre do último ano do Curso de Engenharia Informática, ter comentado com os meus colegas sobre a questão que me levou a escolher este tema, Criptografia. Ainda hoje questiono sobre o verdadeiro motivo porque escolhi este tema. Teria sido pelo desafio? Pela sua complexidade ou pela enorme importância que este tema representa no nosso dia-a-dia?

Afinal o que é a criptografia?

De forma simples, o conceito e objectivo da criptografia é codificar mensagens de forma a assegurar que a integridade da informação seja mantida.

A criptografia é um conjunto de técnicas pensadas para proteger uma determinada informação de modo a que apenas emissor e receptor consigam decifrá-las e lê-las.

O protocolo de criptografia pode ser mais ou menos elaborado e técnicas como essas existem desde a antiguidade, com o primeiro sistema de criptografia conhecido tendo surgido no Egito, cerca de 1.900 anos antes de Cristo.

A criptografia tem um apelo especial para assuntos ligados á guerra, mas comerciantes e governantes também podem ver na criptografia uma saída para evitar que pessoas não autorizadas descubram informações sobre, por exemplo, as suas estratégias. A ideia básica é que este sistema de técnicas cifre uma informação que somente será decifrada por pessoas autorizadas, sem acessos indevidos no caminho.

E perceber como a criptografia funciona é simples: o emissor da mensagem utiliza um protocolo que vai protegê-la, depois é transmitida para o destinatário, que possui uma chave capaz de “resolver” o algoritmo da criptografia e visualizar o seu conteúdo.

Tomemos como exemplo mais conhecido do público em geral, Edward Joseph Snowden. Snowden é um analista de sistemas, ex-administrador de sistemas da Agencia Central de Inteligência (CIA) e antigo empregado da Agência de Segurança Nacional (NSA) dos Estados Unidos da América. Snowden tornou-se conhecido após ter revelado um esquema de espionagem e monitorização global conduzido pela NSA e que desde então passou a defender o uso da criptografia de dados, com o intuito de fugir do controlo de entidades reguladoras, tais como governos e empresas.

Desde a antiguidade até aos dias de hoje é inquestionável a importância que a criptografia tem na troca de informação.

O objectivo deste meu trabalho, visa explicar e demonstrar, a importância da criptografia para a segurança da informação e das comunicações. A primeira instância aponta para a sensibilização da necessidade de se adoptarem estratégias e táticas criptográficas para assegurar, de uma forma eficaz; Integridade, Sigilo e Autenticidade de dados e informação.

Mas admito que a minha ambição levou-me a pesquisas mais realistas, que objectiva a conscientização do mais comum cidadão, quanto à importância da criptografia para a

salvaguarda e inviolabilidade da informação relativas à intimidade, à vida privada, à honra e à imagem das pessoas comuns.

Por outro lado, pretendo demonstrar com este meu projeto, que com o recurso a protocolos de encriptação e comunicação, a rapidez da comunicação e a facilidade de implementação de segurança da informação, que hoje é possível com algumas técnicas e no setor das grandes instituições bancárias e/ou privadas.

Este relatório foi realizado no âmbito do projeto financiado pelo curso de Engenharia Informática, tendo como objetivo o desenvolvimento de uma aplicação de suporte à gestão de recursos humanos. O projeto foi desenvolvido no âmbito do curso de Engenharia Informática, tendo como objetivo o desenvolvimento de uma aplicação de suporte à gestão de recursos humanos.

- O uso das tecnologias de programação (HTML, CSS, JavaScript, PHP) para a criação de uma aplicação web, permitindo a interação com o sistema de gestão de recursos humanos.
- A utilização de ferramentas de desenvolvimento (IDEs) para a criação de uma aplicação web, permitindo a interação com o sistema de gestão de recursos humanos.
- A implementação de testes unitários e de integração para garantir a qualidade do código e a estabilidade do sistema.
- A implementação de testes unitários e de integração para garantir a qualidade do código e a estabilidade do sistema.
- A implementação de testes unitários e de integração para garantir a qualidade do código e a estabilidade do sistema.
- A implementação de testes unitários e de integração para garantir a qualidade do código e a estabilidade do sistema.
- A implementação de testes unitários e de integração para garantir a qualidade do código e a estabilidade do sistema.
- A implementação de testes unitários e de integração para garantir a qualidade do código e a estabilidade do sistema.

## Enquadramento

Este trabalho foi realizado no âmbito do projeto final de curso de Engenharia Informática, tendo como objetivo o desenvolvimento de uma aplicação de suporte empresarial, para o ambiente do Sistema Operativo Windows 8.1/10. Esta aplicação foi desenvolvida de acordo com as seguintes características e requisitos:

- O uso das linguagens de programação; HTML (Javascript, CSS) e C# (permitidas pelas políticas de segurança da informação e administração de sistemas informáticos da instituição bancária),
- O envio de ficheiros de texto diretamente (ponto-a-ponto) a partir de vários postos de trabalho de colaboradores de função técnica “PT cliente” para um posto de trabalho final de um coordenador de área “PT servidor”,
- A encriptação de dados contidos nos ficheiros,
- A descriptação de ficheiros e alteração da nomenclatura de ficheiros de acordo com a transmissão “PT emissor”,
- A visualização de informação e dados de origem da transmissão,
- A transferência, manuseamento e tratamento em matrizes<sup>1</sup> dos dados do ficheiro,
- A visualização dos dados em formato estatístico e com gráficos.

<sup>1</sup> Tabelas com dados separados individualmente e com índice remessivo, nome em inglês arrays.



## Organização do Projeto Global

Este projeto encontra-se organizado em 4 capítulos, descritos da seguinte forma:

**Primeiro capítulo;** a “Introdução” é feita uma descrição do conceito de Criptografia, suas características básicas e diferenças relativamente aos tempos decorridos.

**Segundo capítulo;** refere-se ao “Estado da Arte”, estando este subdividido em quatro (4) subcapítulos, fazendo referência aos diversos métodos de encriptação, com detalhe à evolução dos mesmos ao longo da Era da Informática e menção aos sistemas/métodos que foram usados no projeto.

**Terceiro capítulo;** “Apresentação da aplicação – Cryptoproxy (PT cliente e PT servidor)”, detalha a apresentação estrutural e funcional das versões da aplicação, ou seja, da forma como foram identificados os procedimentos; de tratamento, manuseamento, segurança e envio da informação, tanto na sua componente de técnica como na sua componente interface. Apresenta visualmente a interface em execução.

**Quarto e último capítulo;** consiste na “Conclusão”, é onde são identificadas as principais considerações retiradas ao longo do desenvolvimento deste projeto, resultantes da articulação entre o contexto teórico e desenvolvimento prático da aplicação apresentada.

... e a possibilidade de a partir de agora se trabalhar de forma mais integrada e colaborativa, com a participação de todos os envolvidos no processo de inovação.

4. Oportunidade para a criação de uma cultura de inovação, com a participação de todos os envolvidos no processo de inovação, e a possibilidade de a partir de agora se trabalhar de forma mais integrada e colaborativa, com a participação de todos os envolvidos no processo de inovação.

... e a possibilidade de a partir de agora se trabalhar de forma mais integrada e colaborativa, com a participação de todos os envolvidos no processo de inovação.

... e a possibilidade de a partir de agora se trabalhar de forma mais integrada e colaborativa, com a participação de todos os envolvidos no processo de inovação.

... e a possibilidade de a partir de agora se trabalhar de forma mais integrada e colaborativa, com a participação de todos os envolvidos no processo de inovação.

... e a possibilidade de a partir de agora se trabalhar de forma mais integrada e colaborativa, com a participação de todos os envolvidos no processo de inovação.

... e a possibilidade de a partir de agora se trabalhar de forma mais integrada e colaborativa, com a participação de todos os envolvidos no processo de inovação.

... e a possibilidade de a partir de agora se trabalhar de forma mais integrada e colaborativa, com a participação de todos os envolvidos no processo de inovação.

## 2 - ESTADO DA ARTE

### 2.1 – Introdução à história da criptografia

O ponto forte à volta do tema da criptografia reflete-se na contínua necessidade de se usar e transmitir informação, de a proteger e ao mesmo garantir que o seu conteúdo seja salvaguardado até chegar ao seu destinatário.

A Criptografia, junção de duas palavras gregas kriptós (secreto, escondido) e gráfein (escrita) é o uso de técnicas para transformar texto ou dados legíveis em informação ilegível, que não possa ser compreendida.

Generais, reis e rainhas, durante milénios, procuravam formas eficientes de comunicação, de comandar os seus exércitos e de governar os seus países. A importância de não revelar segredos e estratégias às forças inimigas, motivou o desenvolvimento de códigos e cifras, técnicas para mascarar uma mensagem, possibilitando apenas ao destinatário ler o conteúdo. As nações passaram a criar departamentos para elaborar códigos, por outro lado, surgiram os decifradores de códigos, criando uma corrida armamentista intelectual.

As diversas formas e utilidades dadas aos códigos ao longo do tempo mostram a presença fundamental da matemática na evolução de tal teoria. E evolução é um termo bem apropriado, já que todo código sempre está sob o ataque dos decifradores. Ao desenvolver uma nova arma, relevando a fraqueza de um código, este deixa de ser útil, sendo necessário então, a criação de um novo código que prospera até que decifradores identifiquem suas fraquezas, e assim pelos diante. Ao longo da história, os códigos decidiram o resultado de batalhas.

À medida que a informação se torna cada vez mais valiosa, o processo de codificação de mensagens tem um papel cada vez maior na sociedade.

É comum encontrar relatos na história, de episódios envolvendo os códigos em operações durante guerras, onde criptoanalistas desvendaram o código dos criptógrafos “inimigos”, mas mantiveram tal informação em sigilo, a fim de impedir que novos códigos fossem criados para substituir o decifrado. Assim podiam obter informações extremamente importantes e táticas de defesa e ataque.

De modo a facilitar, dividimos as cifras criptográficas em dois períodos distintos no tempo: a criptografia clássica e a criptografia moderna.

### 2.2 - Criptografia Clássica

Podemos chamar de criptografia clássica o período que vai desde os povos antigos, passando pela Idade Média e chegando até as máquinas eletromecânicas, utilizadas principalmente durante a Segunda Guerra Mundial.

De entre as cifras clássicas mais conhecidas temos o *scytale* espartano, a cifra de César, a cifra de Vigenère e a cifra de Playfair. E como máquina eletromecânica indicamos a não menos famosa *Enigma*.

## Scytale

O método de cifragem com o *scytale* (bastão, no idioma grego) ou *cítala* espartana, consistia em se enrolar uma fita de tecido num bastão de madeira de uma largura específica. A frase a ser cifrada era escrita na fita no comprimento do bastão, desenrolada e enviada disfarçada (e.g. um cinto) e ao chegar ao destino a mesma seria enrolada num bastão da mesma largura para que a mensagem fosse decifrada e assim ser lida corretamente.



Figura 1 - *scytale* (bastão) ou *cítala* espartana

Também era conhecida como bastão de Licurgo, embora alguns estudiosos citem que este tipo de cifra não passa de um mito.

*A partir daqui a criptografia não seria mais a mesma...*

## Cifra de César

Uma das cifras mais conhecidas é a cifra de César, que foi utilizada por Júlio César para o próprio comunicar com as suas tropas durante as guerras que travava.

Esta cifra é bastante simples, consiste na substituição de uma letra do alfabeto pelo seu correspondente de três casas adiante, ou seja, a letra “A” é substituída pela letra “D”, a letra “B” pela letra “E” e assim sucessivamente.

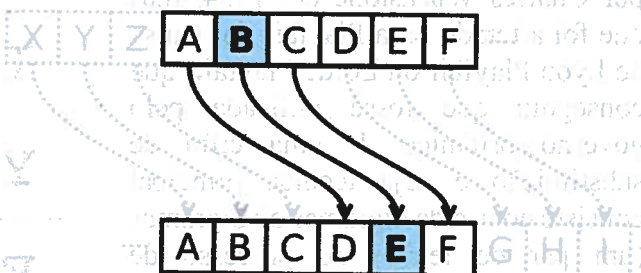


Figura 2 - cifra de César (Júlio César)

Neste caso, o algoritmo da cifra é a troca de uma letra por outra numa determinada posição.

A chave, neste caso o número “3”, é a quantidade de casas a mover.

## Cifra de Vigenère – a cifra indecifrável

A cifra de Vigenère (atribuída incorretamente a Blaise de Vigenère) foi descrita primeiramente pelo italiano Giovan Battista Bellaso, em 1553, em sua obra *La cifra del. Sig. Giovan Batista Bellaso* e por muito tempo foi considerada como *le chiffre indéchiffrable* (a cifra indecifrável) quando, em meados do século XIX, Charles Babbage e Friedrich Kasiski encontraram um método de resolvê-la.

- Example of Vigenère Cipher**
- Plain Text: **ATTACKATDAWN**
  - Key: **LEMON**
  - Plain text: **ATTACKATDAWN**
  - Key: **LEMONLEMONLE**
  - Cipher text: **LXFOPVEFRNHR**

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figura 3 - cifra de Vigenère

O seu processo de cifragem usa o seguinte método: o utilizador cifrará a mensagem com uma chave alfabética; caso a quantidade de caracteres da chave for menor que o tamanho de caracteres da mensagem, a chave será repetida até ambas terem a mesma quantidade de caracteres.

Fazendo uma relação entre as duas (a mensagem e a chave), cada letra da mensagem será cifrada com um alfabeto definido pelo caracter da chave ao qual estará relacionada.

## Cifra Playfair

A cifra Playfair, também conhecida como quadrado de Playfair, foi uma cifra inventada por Charles Wheatstone em 1854, mas que foi alterado para Playfair por causa de Lyon Playfair ou Lorde Playfair, que conseguiu que fosse utilizada pelo governo britânico. É uma cifra de substituição e cuja técnica principal consiste no uso de digramas ou bigramas (um par de letras) no processo de cifragem/decifragem.

O seu método de funcionamento é suportado por uma matriz de formato 5×5, preenchida com as 26 letras do alfabeto latino (como a tabela é um quadrado de 5×5, resultando em 25 células, foi definido unir as letras i e j na mesma célula, onde todo j na mensagem a ser cifrada será substituído por i) e uma palavra-chave ou senha.

I	U	S → T	F	
Y	A	B	C	D
E	G	H	K	L
M	N	O	P	Q
R	V	W ← X	Z	

Figura 4 - cifra Playfair

## A máquina Enigma

Até ao início do século XX, as cifras que foram outrora desenvolvidas podiam ser solucionadas sem a necessidade de uma máquina, bastava tempo e dedicação.

Mas, com o surgir da mecanização, algumas máquinas foram desenvolvidas com o intuito de acelerar tanto o processo de cifragem/decifragem como em dificultar a criptoanálise das mensagens cifradas.

Na tipologia destes equipamentos, o mais conhecido é a máquina Enigma, utilizada pelo exército alemão durante a Segunda Guerra Mundial.

A Enigma identifica-se pela sua apresentação física, um pouco semelhante a uma máquina de escrever, onde ao invés de colocar o resultado num papel, o mesmo era mostrado num painel luminoso com os caracteres do alfabeto.



Figura 5 - máquina Enigma

A chave usada para cifrar/decifrar uma mensagem era configurada num sistema de seletores do tipo de rotores eletromecânicos (3 ou mais) que podiam ser alterados conforme a necessidade para formar a chave.

Foi considerado impossível decifrar uma mensagem cifrada com a Enigma.

A descoberta duma criptoanálise inversa só foi possível devido aos esforços de polacos e ingleses, sendo Alan Turing o mais recordado, como a personagem central na criptoanálise inversa da cifragem da Enigma.

### 2.3 - Criptografia Moderna

Na criptografia moderna, apontamos principalmente às cifras DES, AES e RSA. Este período teve o seu auge na década de 70.

#### Cifra DES (*Data Encryption Standard*)

O DES é um método de cifragem tradicional ("simétrico) desenvolvido nos anos setenta, utiliza uma chave de 56 bits que é aplicada a blocos de dados com 64 bits, o objectivo deste algoritmo é que seja muito difícil calcular a chave  $K$ , mesmo conhecendo o algoritmo DES, uma mensagem cifrada  $C$  e uma mensagem original  $M$ :  $C = \text{DES}(K, M)$ .

O algoritmo é complexo e detalha-se do seguinte modo:

A mensagem de 64 bits é dividida em duas partes de 32 bits cada;

A chave de 56 bits é usada para gerar 16 chaves de 18 bits cada;

É aplicado sucessivamente 16 vezes um algoritmo, usando as chaves geradas.

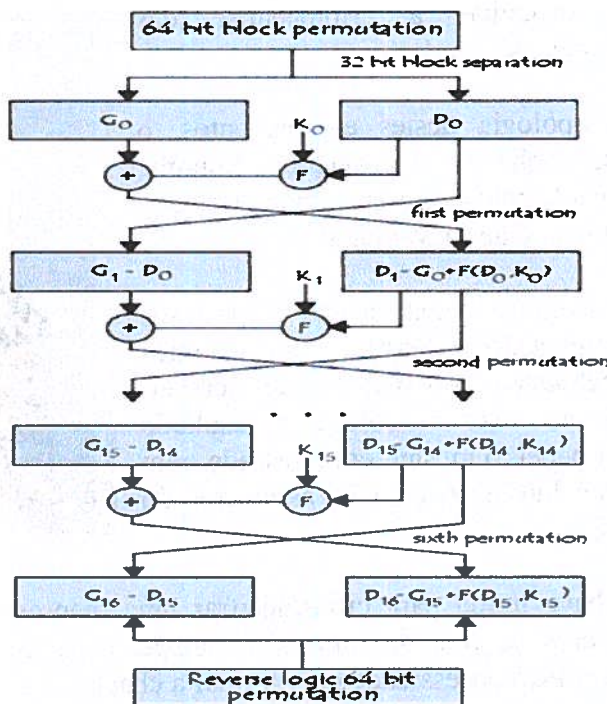


Figura 6 - cifra DES (esquema funcional)

Devido as suas características pequenas alterações na mensagem original provocam grandes alterações na mensagem cifrada, isto dificulta as tentativas de conhecer a chave, mesmo que se possa cifrar aquilo que se pretende.

Note-se que as chaves DES têm 64 bits (8 octetos), mas o algoritmo obriga a que cada octeto da chave seja impar, logo o bit menos significativo de cada octeto não é usado porque tem valor fixo. Em termos reais temos por isso chaves com apenas 56 bits.

## AES (Advanced Encryption Standard)

Principalmente adotada pelo governo dos EUA para proteger informação classificada, o AES recebeu ao longo dos tempos e globalmente a sua aceitação e é usado para proteger informação sensível de várias empresas. A AES pertence à família das cifras conhecidas como cifras de bloco.

Uma cifra de bloco é um algoritmo que encripta dados numa base de blocos. O tamanho de cada bloco é usualmente medido em *bits*. O AES, por exemplo, tem de tamanho 128 *bits*. Resumindo que o AES opera sobre um excerto de texto de 128 *bits* para produzir um texto cifrado de 128 *bits*.

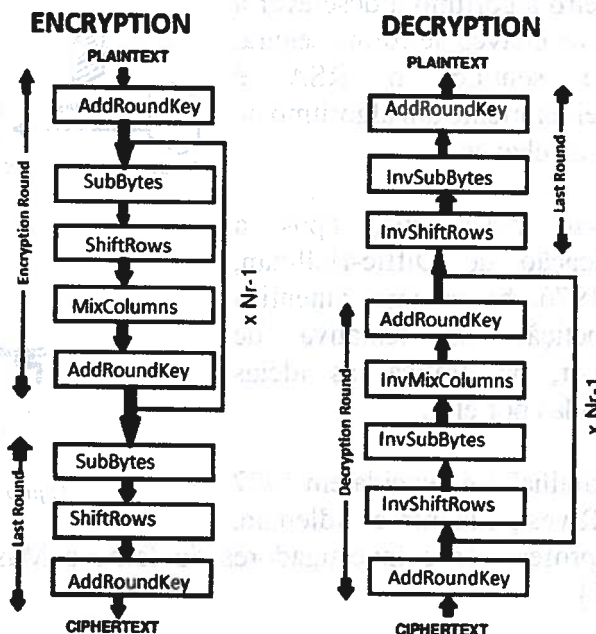


Figura 7 - cifra AES (esquema funcional)

Como a maioria dos algoritmos de encriptação modernos, o AES requer o uso de chaves durante o processo da encriptação e desencriptação.

É suportado por três chaves com comprimentos diferentes: 128 *bits*, 192 *bits* e de 256 *bits*. Quanto maior a chave, mais resistente é a encriptação.

No entanto, no âmbito da velocidade do processo de encriptação e desencriptação, quanto maior o numero de *bits*, maior é o tempo.

Neste algoritmo, as chaves usadas na encriptação, são as mesmas usadas na desencriptação. Assim se diz que usam chaves simétricas.

## Cifra RSA (Rivest, Shamir e Adleman)

A RSA é, em parte, derivado do conceito teórico do algoritmo Diffie-Hellman, que foi o primeiro algoritmo a descrever a troca de chaves de forma segura. Nesse sentido, o RSA é conceitualmente um algoritmo de troca de chaves.

Pode-se dizer que, após a publicação do Diffie-Hellman, em 1976, houve uma autêntica competição na tentativa de traduzir, na prática, as ideias reveladas por eles.

A “batalha” foi vencida em 1977 por Rivest, Shamir e Adleman, três professores e investigadores do famoso Massachusetts Institute of Technology (MIT).

Com base nas ideias de Diffie-Hellman, eles construíram um dos mais poderosos algoritmos criptográficos que o mundo conheceu até a época: o RSA.

Em 1983, a patente do RSA foi aceita, o que, na prática, representou a primeira patente de um algoritmo criptográfico.

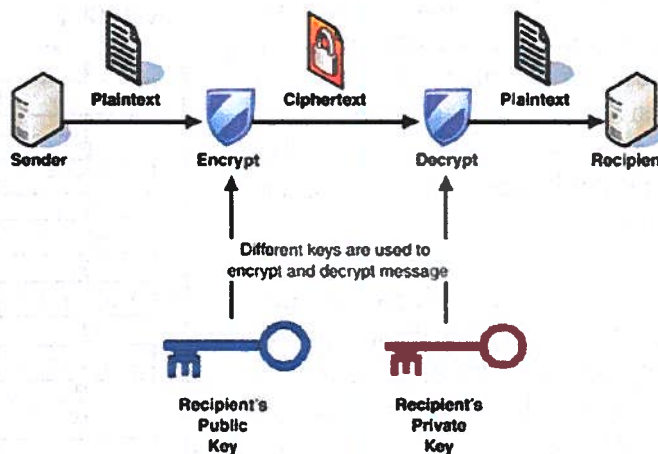


Figura 8 - cifra RSA (esquema funcional)

### 2.3.1 - Resumo

A possibilidade de comunicação entre computadores numa rede, como a internet, trouxe novos desafios para a criptografia.

Por ser relativamente fácil interceptar mensagens inicialmente enviadas por linha telefónica, tornou-se necessário codificá-las, sempre que contenham informações sensíveis, como transações bancárias ou comerciais, ou até mesmo uma compra feita com cartão de crédito.

Imagine-se que uma empresa envia a um banco uma autorização para uma transação de milhões de Euros. Dois problemas surgem imediatamente.

O primeiro é que é preciso proteger a mensagem para que não possa ser lida, mesmo que seja interpretada por um intermediário, ou por alguém ligado ao cibercrime.

Por outro lado, o banco precisa de ter a certeza de que a mensagem foi enviada por um colaborador da empresa, ou seja, como se a mensagem estivesse assinada.

Desta forma, tornou-se necessário inventar novos códigos, que mesmo com a ajuda de um computador, fossem difíceis de decifrar.

A fim de se desenvolver esta forma de criptografia, a ideia seria encontrar uma função de mão única que, como o nome sugere, fosse irreversível.

É como partir um ovo: partir é fácil, mas o impossível é fazer o ovo voltar ao seu estado inicial.

Começou assim um frenético estudo para encontrar uma função matemática apropriada.

## 2.4 - Cryptoproxy

Como provedor de serviços de criptografia vamos considerar como exemplo a Microsoft CryptoAPI.

A Microsoft CryptoAPI fornece um interface seguro para a funcionalidade criptográfica que, é suportada pelo fornecedor de módulos de serviço criptográfico instalados (CSP). Os CSP executam todas as operações criptográficas e gerem as chaves privadas dos CSP que podem ser instaladas em software ou em hardware.

### Fornecedores de serviços criptográficos de Hardware e/ou Software

Os CSP podem ser baseadas em software (CSP software), hardware (CSP hardware) ou um pouco de ambos (CSP software/hardware). A criptografia em CSP hardware e a sua gestão de chaves são mais seguras do que a de criptografia em CSP software e a sua respetiva gestão de chaves, principalmente pelo facto de as operações criptográficas e as chaves privadas estarem isoladas do sistema operativo.

No entanto, os CSP hardware (e.g. smart cards) geralmente armazenam um número limitado de chaves privadas e podem demorar um tempo longo para gerar chaves.

Os CSP software usualmente fornecem mais flexibilidade do que os CSP hardware, mas com alguma segurança a menos. No entanto, os CSP software podem fornecer uma segurança ampla que vai ao encontro de um angulo amplo de necessidades. Normalmente usamos os CSP hardware apenas para aplicações de segurança especiais, tais como executar *login* com um *smart card* (cartão do cidadão online) ou para executar operações num ATM ou TPA.



Figura 9 - cartão com chip (smart card)

Os fabricantes podem desenvolver os CSP hardware e/ou CSP software que suportem uma grande área de operações e tecnologias criptográficas. No entanto, a Microsoft tem de certificar e assinar digitalmente todos os CSP.

Os CSP não irão funcionar no sistema operativo Windows sem a assinatura digital fornecida pela Microsoft.

### 3 – Linguagens de programação

Aqui identifica-se e detalha-se as linguagens de programação usadas no desenvolvimento das aplicações deste projeto; versão “PT *cliente*”, versão “PT *servidor*” e o programa de visualização gráfico.

#### HTML (HyperText Markup Language)

É uma linguagem de marcação utilizada para desenvolvimento de *websites*. A sua criação data de 1991, por Tim Berners-Lee, no CERN (European Organization for Nuclear Research) na suíça. Inicialmente o HTML foi projetado para interligar instituições de pesquisa próximas umas das outras e compartilhar documentos com facilidade.

Em 1992, foi disponibilizada a biblioteca de desenvolvimento WWW (World Wide Web), uma rede de alcance mundial, que junto com o HTML proporcionou o uso em escala mundial da WEB. O HTML é uma linguagem de marcação (TAGS), constituída de códigos que delimitam conteúdos específicos, segundo uma sintaxe própria.

O HTML foi a primeira linguagem de nível mundial, no entanto não era a única. Existiam muitas outras linguagens destinadas á criação de páginas da web, porém o HTML ainda prevalece. Atualmente já é possível integrar varias linguagens na mesma página da Web.

Para a criação e edição de código HTML, é apenas necessário um editor de texto comum, como bloco de notas. Para testar os códigos, basta salvar o arquivo num ficheiro com extensão (.html) e executar.

#### Amostra do código HTML usado:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <link href="CSS\ProjectoISTEC.css" type="text/css" rel="stylesheet"></link>
  <title>A.A.C. Tempos/Quantidades</title>

  <!-- JQuery js -->
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>

  <!-- <script src = "d://FlowCount//JS//jquery-migrate-1.9.1.min.js"></script> -->
  <script src="https://code.highcharts.com/highcharts.src.js"></script>

  <script src="/JS/highcharts.js" type="text/javascript"></script>

  <!-- HIGHCHARTS js -->
  <script src="/JS/exporting.js" type="text/javascript"></script>

  <!-- HIGHCHARTS livreria de exportação js -->
  <script src="/JS/HC.js" type="text/javascript"></script>

  <!-- HIGHCHARTS Construção de gráficos js -->
  <script src="/JS/fileOpen.js" type="text/javascript"></script>

  <!-- importação de ficheiro dados js -->
```

```

<script src="/JS/inicializaMatrizBD.js" type="text/javascript"></script>

<!-- importação de ficheiro dados js -->
<script type="text/javascript">

    inicializaMatrizes();
    mediaTempoDiarioCLVL();
    mediaDuracaoTotal();
    medias();

</script>

</head>

<body>

    <header>

<h1>AUTORIZAÇÃO DE ABERTURA DE CONTA</h1>
<h2>Estatística de validações e tempos de validação</h2>

</header>

    <section>
<table>
    <tr>
        <td colspan="6"><center><th colspan = "6">RESUMO A.A.C.</th></center>
    </tr>
    <tr>
        <th>Total de Validações</th>
        <th>Total de Tempo</th>
        <th>Total de Tempo Médio</th>
        <th>Média Diária de Clientes Validados</th>
        <th>Média Mensal de Clientes Validados</th>
        <th>Tempo Médio Diário de Clientes Validados</th>
    </tr>
    <tr>
        <td id = "totalAAC">totalAAC</td>
        <td id = "totalTempoAAC">totalTempo</td>
        <td id = "totalTempoMedioAAC">totalMedia</td>
        <td id = "MediaDiariaCLVL"></td>
        <td id = "MediaMensalCLVL"></td>
        <td id = "TempoMedioDiarioCLVL">teste</td>
    </tr>
    <tr>
        <td colspan="6"><center></center>
    </tr>
    <tr>
        <th>Clientes Validados</th>
        <th>Para Analisar</th>
        <th>Por Verificar</th>
        <th>Total de dias com Clientes Validados</th>
        <th>Total de meses com Clientes Validados</th>
    </tr>
    <tr>
        <td id = "totalCLVL">totalCLVL</td>
        <td id = "totalCLparaAnalisar">totalCLparaAnalisar</td>
        <td id = "totalCLporVerificar">totalCLporVerificar</td>
        <td id = "totalDiasCLVL">totalDiasCLVL</td>
        <td id = "totalMesesCLVL">totalMesesCLVL</td>
    </tr>
</table>

```

```

<footer>
Gráficos Demonstrativos
</footer>
<div id = "grafico-CLVL-Diario">
</div>
<p>
</p>

<div id = "grafico-TempoMedioCLVL-Diario">
</div>
<p>
</p>

<div id = "grafico-CLVL-Mensal">
</div>
<p>
</p>

</section>

<footer>
Direitos Reservados - Tiago ADM Correia @ CGD 2017 - Projecto ISTEC
</footer>

<script>
    document.getElementById("totalAAC").innerHTML = totalValidacoes.toString();
    document.getElementById("totalTempoAAC").innerHTML = totalTempo.toString();
    document.getElementById("totalTempoMedioAAC").innerHTML =
totalMediaTempo.toString();
    document.getElementById("totalCLVL").innerHTML = totalCLVL_HTML.toString();
    document.getElementById("totalCLparaAnalisar").innerHTML =
totalCLparaAnalisar_HTML.toString();
    document.getElementById("totalCLporVerificar").innerHTML =
totalCLporVerificar_HTML.toString();
    document.getElementById("MediaDiariaCLVL").innerHTML =
MediaDiariaCLVL_HTML.toString();
    document.getElementById("MediaMensalCLVL").innerHTML =
MediaMensalCLVL_HTML.toString();
    document.getElementById("TempoMedioDiarioCLVL").innerHTML =
TempoMedioDiarioCLVL_HTML.toString();
    document.getElementById("totalDiasCLVL").innerHTML =
totalDiasCLVL_valores_HC.toString();
    document.getElementById("totalMesesCLVL").innerHTML =
totalMesesCLVL_valores_HC.toString();
</script>

</body>
</html>

```

## CSS (*Cascading Style Sheets*)

É uma linguagem que, numa estrutura de folhas de estilo em cascata, permite definir como um documento escrito em linguagem HTML é apresentado visualmente. Esta apresentação, ou formatação de estilos, tem como alguns elementos; as fontes, a cor e tamanho das letras, espaçamento do texto, estrutura visual da página de internet, entre outros.

Ao invés de se repetir estas definições no documento em HTML ou alterar o documento constantemente, existe uma opção de se criar um *link* para um ficheiro de extensão (.css) permitindo assim executar alterações num único local.

### Exemplo de código CSS:

```
header {
  background-color:#7D8686;
  color:white;
  text-align:center;
  padding:5px;
}
nav {
  line-height:30px;
  background-color:#D6E3E3;
  height:300px;
  width:100px;
  float:left;
  padding:5px;
}
section {
  width: auto;
  float: none;
  padding: 15px;
}
footer {
  background-color:black;
  color:white;
  clear:both;
  text-align:center;
  padding:5px;
}
table, th, td {
  padding: 5px;
  border-spacing: 15px;
  text-align: center;
  border: 5px double black;
}
th, td, div {
  border: 2px groove black;
}
```

## JavaScript

É uma linguagem de programação de *Scripting* e orientada a objetos, criada por Brendan Eich, nos anos 90 através da Netscape<sup>1</sup> e em parceria com a Sun Microsystem<sup>2</sup>.

Foi originalmente desenvolvida como parte integrante dos exploradores de internet (*internet browsers*) para que os *scripts* fossem executados do lado do cliente e interagissem com o utilizador, sem a necessidade de passar por um servidor.

<sup>1</sup> Netscape Communications Corporation é uma empresa dos EUA que desenvolveu o browser Netscape

<sup>2</sup> Empresa dos EUA fabricante de hardware e que foi adquirida pela Oracle em 2009

**Exemplo de código JavaScript:**

```

function mediaTempoDiarioCLVL()
{
    var matrizTempoDiario = [];
    var matrizTempoMedioDiario = [];
    var tempMatriz = [];
    var x = 0;
    var y = 0;
    var tempDataHora;
    var hora = 0;
    var minuto = 0;
    var segundo = 0;
    var totalSegundos = 0;
    var segundos = 0;
    var media = 0;
    var N = 0;
    var tempSegundos = 0;

    tempMatriz = arrayBD.slice();

    for (x = 1; x < tempMatriz.length; x++)
    {
        if (tempMatriz[x][9] == VALIDACAO[0])
        {
            tempDataHora = tempMatriz[x][0].split(/-| /); // Divide data com
            hora inicio // Retira hora de
            tempDataHora.pop();
            inicio
            tempDataHora.push(tempMatriz[x][2].split(/:/)); // Adiciona duração
            validação

            hora = parseInt(tempDataHora[3][0]);
            minuto = parseInt(tempDataHora[3][1]);
            segundo = parseInt(tempDataHora[3][2]);

            totalSegundos = (hora * 3600 + minuto * 60 + segundo); // Converte duração
            em segundos
            tempDataHora.pop(); // Remove duração
            validação
            tempDataHora.push(totalSegundos); // Adiciona duração
            em segundos

            matrizTempoDiario.push(tempDataHora);
        }
    }

    for (x = 0, y = 0; x < matrizTempoDiario.length; x++)
    {
        if (x == 0)
        {
            matrizTempoMedioDiario = matrizTempoDiario.slice(0,1);
            N++;
            document.write("Media de Tempo Diario para CL.VL.<br>");
        } else
        {
            if (matrizTempoDiario[x][0] === matrizTempoMedioDiario[y][0]) //
            Confirma Ano
            {
                if (matrizTempoDiario[x][1] === matrizTempoMedioDiario[y][1]) //
                Confirma Mês
                {
                    if (matrizTempoDiario[x][2] ===
                    matrizTempoMedioDiario[y][2]) // Confirma Dia
                    {
                        segundos = parseInt(matrizTempoDiario[x][3]);
                        matrizTempoMedioDiario[y][3] += segundos;
                        N++;
                    } else
                    {
                        tempSegundos = (matrizTempoMedioDiario[y][3] /
                        N).toFixed(0);
                        matrizTempoMedioDiario[y][3] = tempSegundos;
                        matrizTempoMedioDiario.push(matrizTempoDiario[x++]);
                    }
                }
            }
        }
    }
}

```

```

        y++;
        x--;
        N = 1;
    }
} else
{
    tempSegundos = (matrizTempoMedioDiario[y][3] /
N).toFixed(0);

    matrizTempoMedioDiario[y][3] = tempSegundos;
    matrizTempoMedioDiario.push(matrizTempoDiario[x++]);
    y++;
    x--;
    N = 1;
}
} else
{
    tempSegundos = (matrizTempoMedioDiario[y][3] / N).toFixed(0);
    matrizTempoMedioDiario[y][3] = tempSegundos;
    matrizTempoMedioDiario.push(matrizTempoDiario[x++]);
    y++;
    x--;
    N = 1;
}
}
}
for (y = 0; y < matrizTempoMedioDiario.length; y++)
{
    document.write(matrizTempoMedioDiario[y] + "<br>");
}
}

```

## C# (C Sharp)

É uma linguagem de programação com suporte na plataforma “.NET”. É uma linguagem orientada por objetos (OOP), suporta a criação de métodos, tipos, classes e estruturas leves. O seu desenvolvimento teve como âmbito a utilização do conhecimento das linguagens C e C++ (C *plusplus*), mas garantindo constantemente uma performance e segurança superior.

### Exemplo de código C# (versão “PT cliente”):

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
// adicionadas
using System.Threading;
using System.Net;
using System.Net.Sockets;
using System.IO;
using System.Diagnostics;
using TCP_Simple_Client.BO;

namespace TCP_Simple_Client
{
    public partial class Form1 : Form
    {
        String[] cifrada;
        String filename_cifrada;
    }
}

```

```

public Form1()
{
    InitializeComponent();
}

private void btnBrowse_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();

    openFileDialog.InitialDirectory = "d:\\";
    openFileDialog.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
    openFileDialog.FilterIndex = 1;
    openFileDialog.RestoreDirectory = true;
    openFileDialog.ShowDialog();
    tbFilename.Text = openFileDialog.FileName;

    Cripto _encriptar = new Cripto();

    string[] lines = File.ReadAllLines(tbFilename.Text);

    cifrada = new String[lines.Length];

    for (int i = 0; i < lines.Length; i++)
    {
        if (!lines[i].Equals(null))
        {
            String cript_Line = _encriptar._caesar(lines[i], true);

            cifrada[i] = cript_Line;
            //MessageBox.Show(lines[i] + " : " + cifrada[i]);
        }
    }
    filename_cifrada = tbFilename.Text + "_cifra";

    File.WriteAllLines(filename_cifrada, cifrada); //
}

private void btnSend_Click(object sender, EventArgs e)
{
    // Stopwatch object constructor
    Stopwatch _cronometro = new Stopwatch();

    // Starts the count
    _cronometro.Start();

    Stream fileStream = File.OpenRead(filename_cifrada);
    // Allocate memory space for the file

    byte[] fileBuffer = new byte[fileStream.Length]; // array byte

    //byte[] cifradatoByte = Convert.FromBase64String(cifrada.ToString());

    fileStream.Read(fileBuffer, 0, (int)fileStream.Length);

    // Open a TCP/IP Connection and send the data
    TcpClient clientSocket = new TcpClient(tbServer.Text, 8080);
    NetworkStream networkStream = clientSocket.GetStream();
    networkStream.Write(fileBuffer, 0, fileBuffer.GetLength(0));
    networkStream.Close();

    // Stops the count
    _cronometro.Stop();

    // Get the elapsed time as a TimeSpan value.
    TimeSpan _tempodecorrido = _cronometro.Elapsed;

    // Format and display the TimeSpan value.
    string elapsedTime = String.Format("{0:00}:{1:00}:{2:00}.{3:00}",
    _tempodecorrido.Hours, _tempodecorrido.Minutes,
    _tempodecorrido.Seconds, _tempodecorrido.Milliseconds / 10);

    //Mostra caixa de texto com tempo decorrido
    MessageBox.Show("RunTime: " + elapsedTime);
}
}
}

```

**Exemplo de código C# (método de criptografia):**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace TCP_Simple_Client.BO
{
    partial class Cripto
    {
        public Cripto()
        {
        }

        private string _letragrande(string amostra) // CONVERSÃO PARA LETRA GRANDE
        {
            string resultado = amostra.ToUpper();

            return resultado;
        }

        private int _codigoascii(char amostra) // RETORNA CODIGO ASCII
        {
            int codigoascii = (int)amostra;
            return Convert.ToByte(codigoascii);
        }

        private string _metodo(string amostra, bool metodo) // MÉTODO "CAESAR"
        {
            string cifra = null;

            if (metodo == true)
            {
                for (int i = 0; i < amostra.Length; i++)
                {
                    int codigoascii = _codigoascii(amostra[i]) + 3; //ENCRIPITAR
                    CONTEUDO
                    cifra += "" + Convert.ToChar(codigoascii);
                }
            }
            else if (metodo == false)
            {
                for (int i = 0; i < amostra.Length; i++)
                {
                    int codigoascii = _codigoascii(amostra[i]) - 3; //DECRIPITAR
                    CONTEUDO
                    cifra += "" + Convert.ToChar(codigoascii);
                }
            }
            return _letragrande(cifra);
        }

        public string _caesar(string amostra, bool metodo) // INICIA ENCRIPTAÇÃO
        {
            string cifra_caesar = _metodo(amostra, metodo);

            return (cifra_caesar);
        }
    }
}

```

**Exemplo de código C# (versão "PT servidor"):**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
// adicionadas
using System.Threading;
using System.Net;
using System.Net.Sockets;
using System.IO;
using System.Collections;
using TCP_Simple_Client.BO;

namespace TCP_Server
{
    public partial class Form1 : Form
    {
        public string nomeHOST;
        private ArrayList nSockets;

        public Form1()
        {
            InitializeComponent();
            this.Text = "PT servidor - Cryptoproxy";
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            string str_dir_padrao = "d:\\RCC\\";
            txb_dir_select.Text = str_dir_padrao;

            // Adicionado
            IPHostEntry IPHost = Dns.GetHostEntry(Dns.GetHostName());
            listBox1.Items.Add("Hostname: " + IPHost.HostName.ToString());
            foreach (IPAddress address in IPHost.AddressList)
            {
                string str = address.ToString();
                if (str.Length <= 13 && str.Length <= 15)
                {
                    listBox1.Items.Add("IP: " + address.ToString());
                }
                else
                {
                    //listBox1.Items.Add("MAC: " + address.ToString());
                }
            }

            nSockets = new ArrayList();
            Thread thdListener = new Thread(new ThreadStart(listenerThread));
            thdListener.Start();
        }

        public void listenerThread()
        {
            TcpListener tcpListener = new TcpListener(IPAddress.Any, 8080);
            tcpListener.Start();
            while (true)
            {
                Socket handlerSocket = tcpListener.AcceptSocket();
                if (handlerSocket.Connected)
                {
                    Control.CheckForIllegalCrossThreadCalls = false;
                    lbConnections.Items.Add(
                        handlerSocket.RemoteEndPoint.ToString() + " ->> conexão IP
recebida.");
                    lock (this)
                    {
                        nSockets.Add(handlerSocket);
                    }
                }
            }
        }
    }
}

```

```

        ThreadStart thdstHandler = new
        ThreadStart(handlerThread);
        Thread thdHandler = new Thread(thdstHandler);
        thdHandler.Start();
    }
}

public void handlerThread()
{
    Socket handlerSocket = (Socket)nSockets[nSockets.Count - 1];
    NetworkStream networkStream = new NetworkStream(handlerSocket);
    int thisRead = 0;
    int blockSize = 1024;
    Byte[] dataByte = new Byte[blockSize];
    string linha_txt = string.Empty;
    lock (this)
    {
        // UM PROCESSO ACEDIDO DE CADA VEZ
        var FName = handlerSocket.RemoteEndPoint.ToString().Substring(0,
handlerSocket.RemoteEndPoint.ToString().Length - 6) +
        "-" + DateTime.Now.ToString("dd_MM_yyyy-hhh-mm-ss");

        Stream fileStream = File.OpenWrite(txb_dir_select.Text.ToString() + "\\\"
+ FName + ".txt_cifra");
        while (true)
        {
            thisRead = networkStream.Read(dataByte, 0, blockSize);
            linha_txt = System.Text.Encoding.ASCII.GetString(dataByte);
            listBox_dados_encryptados.Items.Add(linha_txt);
            fileStream.Write(dataByte, 0, thisRead);
            if (thisRead == 0) break;
        }
        lbConnections.Items.Add(DateTime.Now.ToLocalTime() + " ->> " + "Ficheiro
recebido");
        lbConnections.Items.Add(FName + ".txt" + " ->> " + "Ficheiro gravado");
        fileStream.Close();

        // INICIA O DESENCRIPTAR DO FICHEIRO .TXT_CIFRA RECEBIDO NUM FICHEIRO .TXT COM
CONTEUDO EM CLARO

        Cripto _encryptar = new Cripto();
        string[] lines = File.ReadAllLines(txb_dir_select.Text + "\\\" + FName +
".txt_cifra");
        string[] em_claro = new String[lines.Length];
        for (int i = 0; i < lines.Length; i++)
        {
            if (!lines[i].Equals(null))
            {
                String cript_Line = _encryptar._caesar(lines[i], false);

                em_claro[i] = cript_Line;
                //MessageBox.Show(lines[i] + " : " + em_claro[i]);
            }
        }
        File.WriteAllLines(txb_dir_select.Text + "\\\" + FName + ".txt",
em_claro);

        string[] files = Directory.GetFiles(txb_dir_select.Text);

        //MessageBox.Show("Files found: " + files.Length.ToString(), "Message");
        listBox1.Items.Add("Ficheiro existentes na pasta: " +
files.Length.ToString());
        var contador = 1;
        foreach (string ficheiros in files)
        {
            listBox1.Items.Add("Ficheiro: " + contador.ToString() + " - " +
ficheiros.Substring(3));
            contador++;
        }
        listBox1.SelectedIndex = listBox1.Items.Count - 1;
    }
    handlerSocket = null;
}
}

```

```

private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    //lblStatus.Text = listBox1.SelectedItem.ToString();
}

public void bt_fechar_app_Click(object sender, EventArgs e)
{
    Environment.Exit(0);
    //Application.Exit();
}

private void bt_save_local_Click(object sender, EventArgs e)
{
    using (var fbd = new FolderBrowserDialog())
    {
        DialogResult result = fbd.ShowDialog();

        if (result == DialogResult.OK &&
!string.IsNullOrEmpty(fbd.SelectedPath))
        {
            string[] files = Directory.GetFiles(fbd.SelectedPath);

            //MessageBox.Show("Files found: " + files.Length.ToString(),
"Message");
            listBox1.Items.Add("Caminho selecionado: " +
fbd.SelectedPath.ToString());
            listBox1.Items.Add("STATUS do Caminho selecionado: " +
DialogResult.OK.ToString());
            listBox1.Items.Add("Ficheiro existentes na pasta: " +
files.Length.ToString());
            var contador = 1;
            foreach (string ficheiros in files)
            {
                listBox1.Items.Add("Ficheiro: " + contador.ToString() + " - " +
ficheiros.Substring(3));
                contador++;
            }
            txb_dir_select.Text = fbd.SelectedPath;
            listBox1.SelectedIndex = listBox1.Items.Count - 1;
        }
    }
}

private void txb_dir_padrao_TextChanged(object sender, EventArgs e)
{
}
}
}

```

### 3.1 - Resumo

Inicialmente, a programação era feita através de código binário (0 e 1 - Assembler). Este era um processo bastante complicado e era muito fácil cometer erros.

Para ajudar a corrigir estas situações, começaram a ser desenvolvidas as linguagens de programação (BASIC, C), que viriam a substituir as funções do código de máquina.

Em seguida surgiram as linguagens orientadas a objetos. Nelas, os dados e as rotinas para manipulá-los são mantidos numa unidade chamada objeto.

A linguagem de programação é o método que o programador utiliza para dar instruções a um computador. A linguagem permite ao programador exprimir exatamente onde quer que o computador atue, tornando-lhe assim a “vida” mais fácil.

Com isto, traz uma maior produtividade, ajudando-os a expressar as suas intenções mais facilmente, tornando-se assim uma ferramenta importante para se escrever programas mais organizados, em menor tempo.

As linguagens de programação acabam por tornar os programas menos dependentes dos computadores. Pois os programas escritos em linguagens de programação são traduzidos para o código da máquina do computador onde será executado, em vez de ser diretamente executado.

Neste mundo “quase” todo tecnológico, qualquer iniciante em programação faz sempre esta pergunta e direcionada a quem já programa;

“Qual é a melhor linguagem de programação para aprender ou arranjar emprego?”

Na resposta a esta questão podemos apontar a permanente necessidade em se suportar os equipamentos e serviços tecnológicos atuais. Como referência, identificámos para o ano de 2017, uma lista com as 10 linguagens de programação mais usadas:























Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

Figura 10 - Top Ten Languages 2017

#### 4 – Protocolo de Comunicação

O protocolo usado neste projeto garante uma comunicação fiável, robusta e rápida, permitindo que os dados transportados nele sejam enviados numa sequência correta e verificados na possibilidade de erros de pacotes de dados.

Caso a transmissão de dados não necessite de verificação de erros nos pacotes de dados, o protocolo de comunicação UDP seria o mais indicado. O mesmo é usado, mas não obrigatório, nas comunicações de voz e outras finalidades que não necessitem de verificação de erros sobre a transmissão dos pacotes de dados.

### **TCP/IP (*Transport Control Protocol / Internet Protocol*)**

É um protocolo de comunicação que está definido na camada de rede OSI (Open System Interconnection - layer 3) e que permite fornecer endereçamento, roteamento e outras funções, numa rede informática. TCP é o primeiro protocolo da camada de transporte e é responsável pela ligação, gestão e fiabilidade da informação transmitida entre os softwares nos equipamentos informáticos. IP fornece comunicação de dados ponto-a-ponto, especificando como os dados são preparados, endereçados, transmitidos e recebidos.

#### **Exemplo de código C#, com o protocolo TCP/IP:**

```
// CRIAR UMA LIGAÇÃO TCP/IP E ENVIAR DADOS

TcpClient clientSocket = new TcpClient(tbServer.Text, 8080); // CRIAR SOCKET COM IP E
PORTA 8080

NetworkStream networkStream = clientSocket.GetStream(); // CRIAR OBJETO PARA A STREAM DE
TRANSMISSÃO

networkStream.Write(fileBuffer, 0, fileBuffer.GetLength(0)); // ESCREVER O CONTEUDO COM
O TAMANHO

networkStream.Close(); // FECHAR A STREAM DE TRANSMISSÃO
```



### 3 - Apresentação da aplicação – Cryptoproxy (PT cliente e PT servidor)

Neste capítulo vou apresentar a aplicação desenvolvida para este projeto, onde o seu objetivo é o de enviar e receber dados cifrados e apresenta-los em modo visual gráfico.

#### 3.1 – Características da Aplicação

O desenvolvimento da aplicação teria de respeitar ao máximo um conjunto de definições de acordo com a política de segurança implementada nos sistemas de informação da instituição bancária:

- Programada em linguagem que não usasse um compilador externo;
- Protocolo TCP/IP, sendo o único protocolo em que as portas lógicas estavam disponíveis para uso;
- A encriptação do conteúdo do ficheiro a enviar;
- Salvaguarda com nomenclatura identificativa do PT (Posto de Trabalho) emissor;
- A desencriptação desse mesmo conteúdo;
- O tratamento, manuseamento e visualização dos dados em modo gráfico e estatístico.

### 3.2 – Funcionalidades da Aplicação

#### A versão “PT Cliente”

A “PT Cliente” tem como objetivo o envio de ficheiros de texto, que são um “*export*” de dados de uma aplicação bancária.

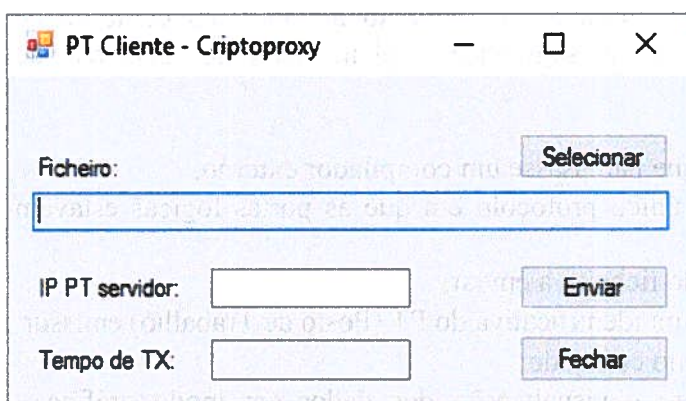


Figura 11 - versão "PT cliente"

Em primeiro lugar, faremos a seleção do ficheiro de texto a enviar.

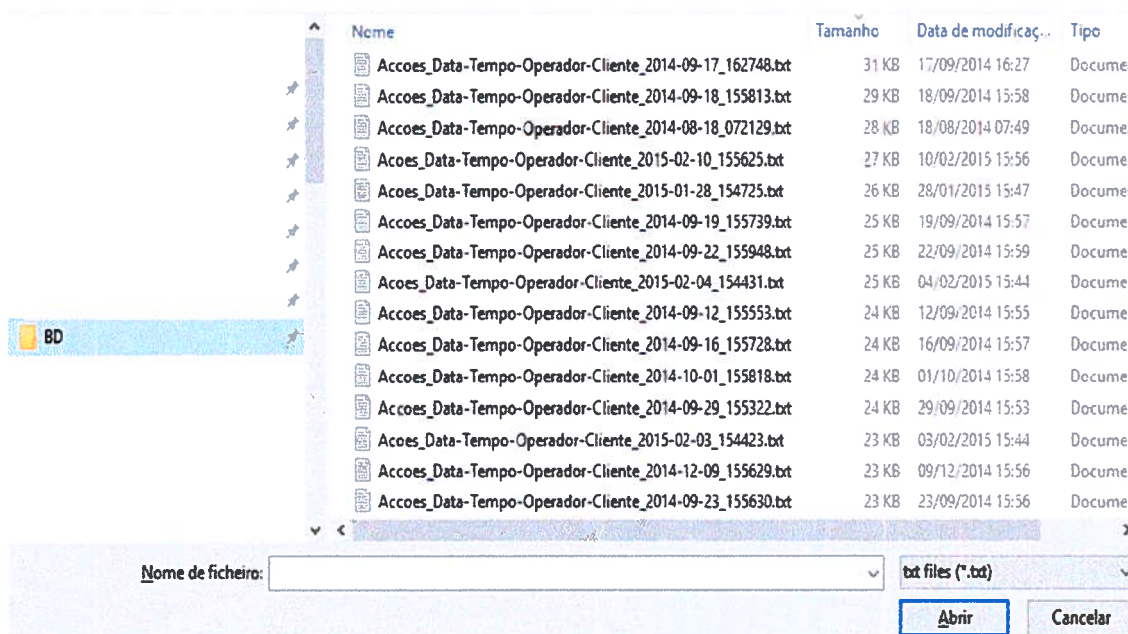


Figura 12 - selecionar ficheiro de texto

O segundo passo será digitar o endereço IP do “PT servidor” e clicar em enviar. Neste momento a aplicação fará uma tentativa de ligação à aplicação versão “PT servidor”, que caso seja estabelecida, o conteúdo será encriptado e transmitido.

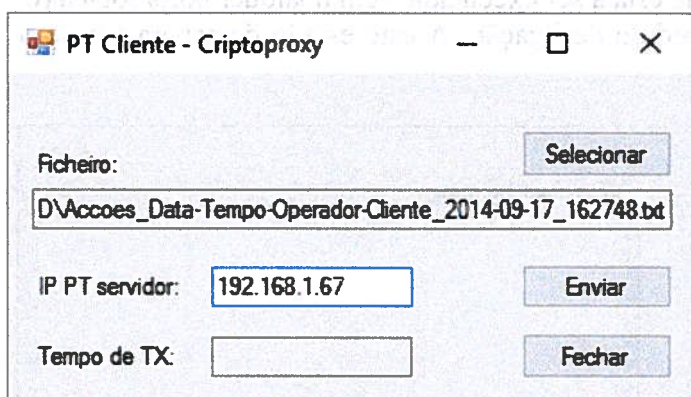


Figura 13 - introduzir IP "PT servidor"

Após sucesso na transmissão, é informado ao utilizador o tempo de duração de ligação na transferência dos dados.

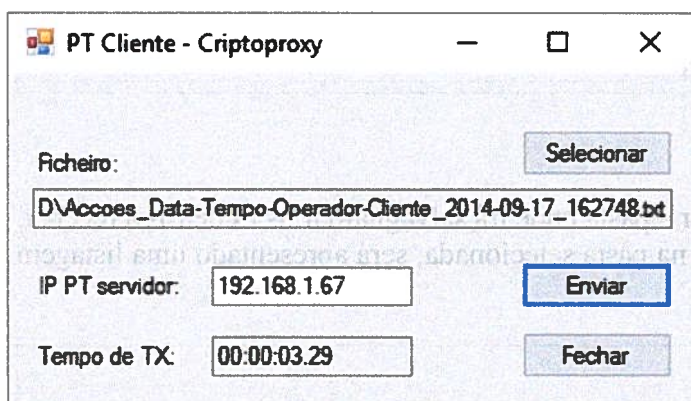


Figura 14 - envio e tempo de transmissão

## A versão “PT servidor”

Do lado da versão “PT servidor”, a ligação TCP está a aguardar um pedido de ligação no IP (e. g. 192.168.1.67) do PC onde está a ser executada e em qualquer porta lógica (e. g. porta 8080) em que receba um pedido de ligação. A este estado de espera usa-se o termo de; modo de “escuta”.

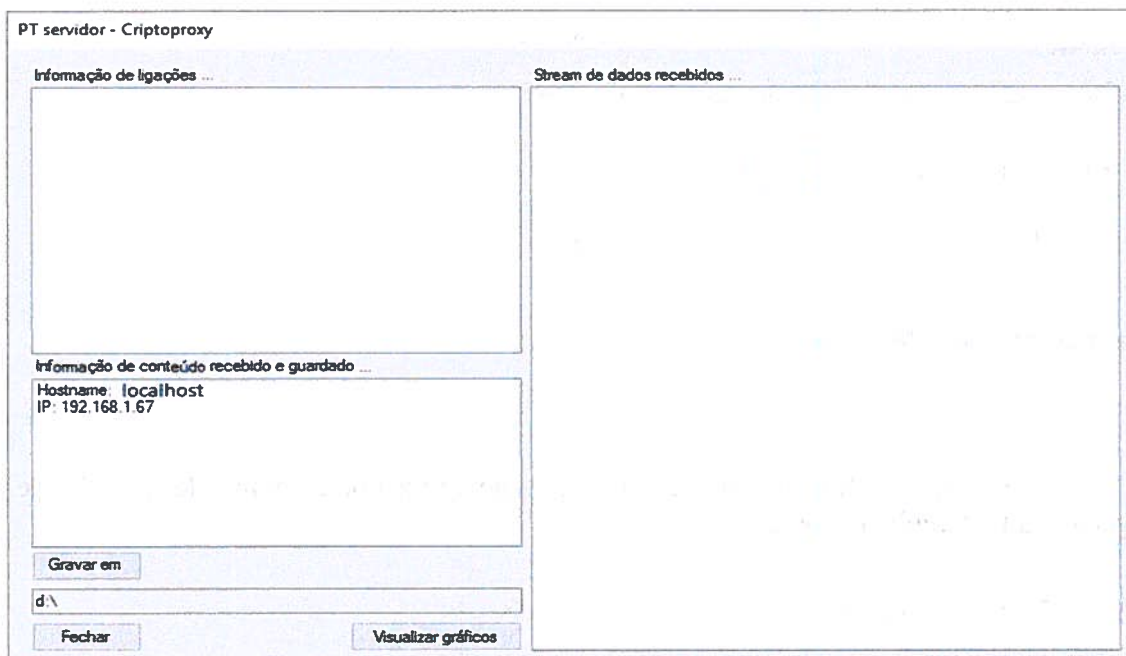


Figura 15 - versão "PT servidor"

O primeiro passo será seleccionar a pasta onde irá salvaguardar os ficheiros a receber. Caso existam inicialmente ficheiros na pasta seleccionada, será apresentado uma listagem com o conteúdo da mesma.

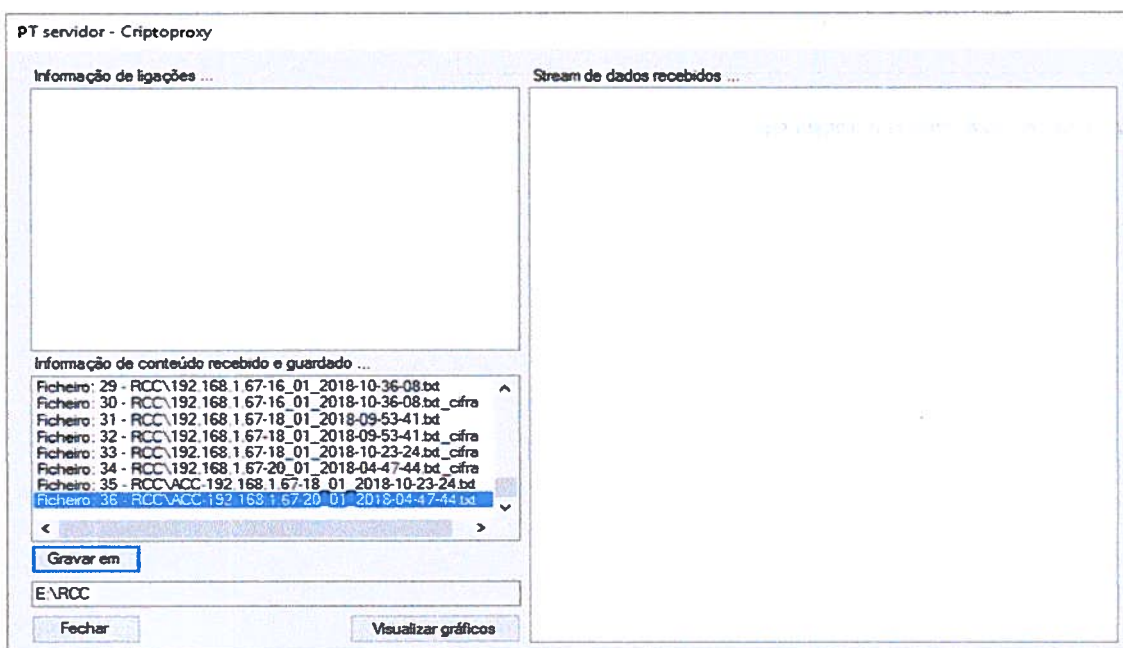


Figura 16 - seleccionar pasta de destino

Ao detetar e estabelecer uma ligação, o “PT servidor” guardará o conteúdo encriptado recebido num ficheiro de texto, com a denominação no seguinte modo:

IP “PT cliente” mais data e hora da transmissão concluída mais \_cifra na extensão (e. g. 192.168.1.67-15\_01\_2018-09-58-23.txt\_cifra).

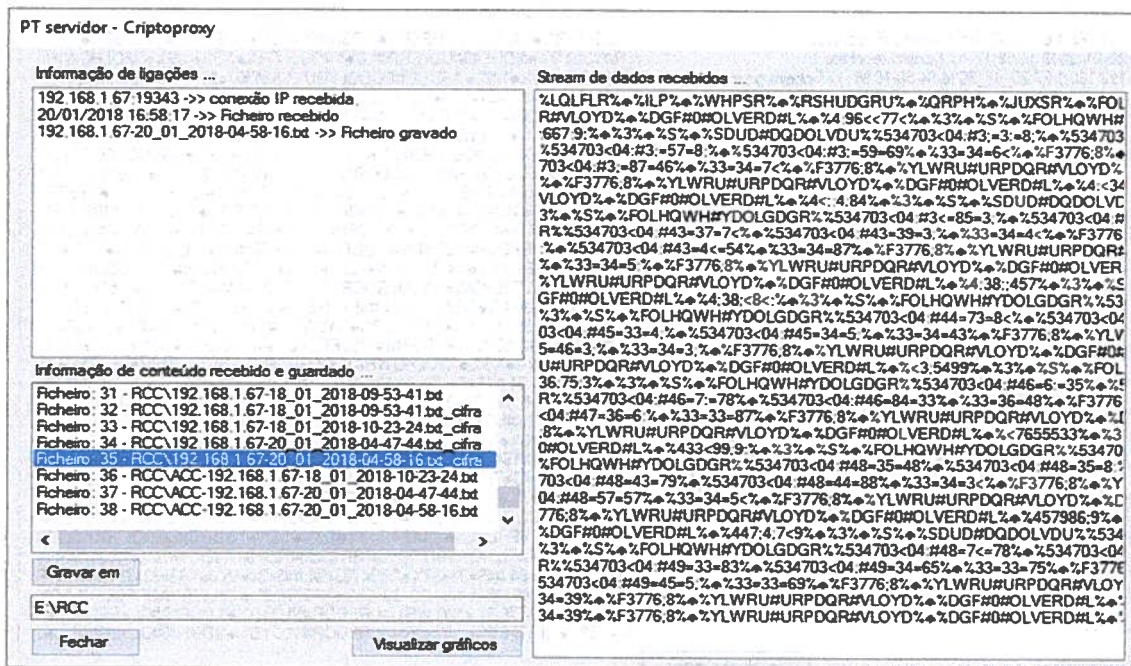


Figura 17 - ficheiro recebido encriptado

Finda a recepção e salvaguarda do ficheiro de texto com conteúdo cifrado, o mesmo será lido novamente pela aplicação que fará a descriptação e novo salvaguardar em novo ficheiro e com o mesmo nome excetuando o texto (\_cifra) na extensão. (e. g. 192.168.1.67-15\_01\_2018-09-58-23.txt).

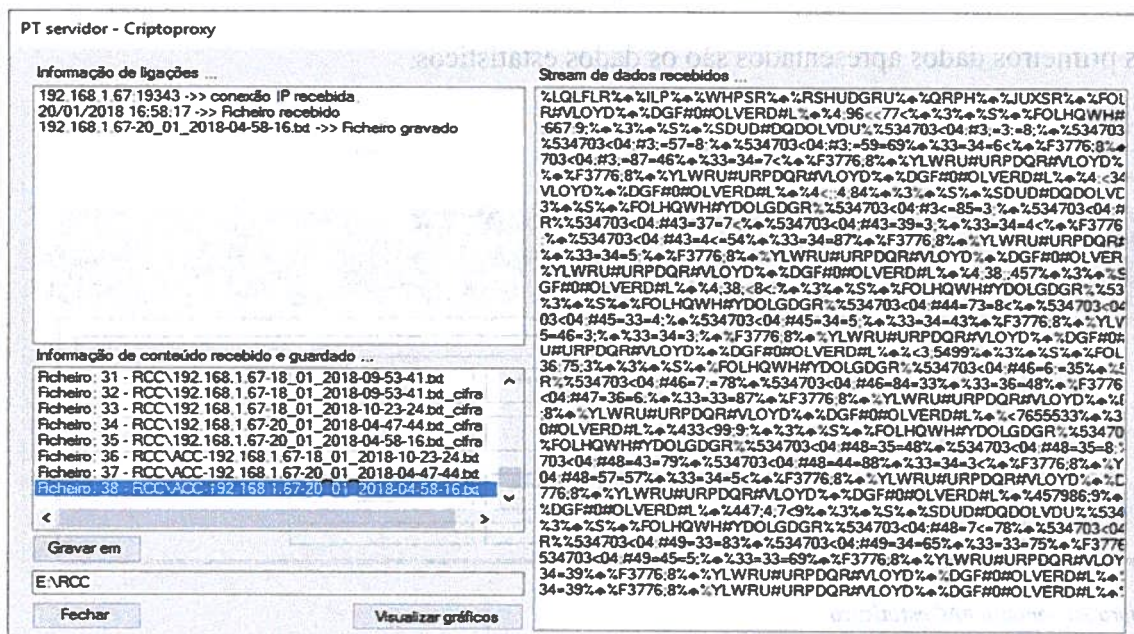


Figura 18 - ficheiro descriptado

O utilizador após a receção e descriptação do ficheiro de texto, terá de o seleccionar no quadro de informação de ficheiros em pasta e clicar “Visualizar gráficos” para proceder à fase final.

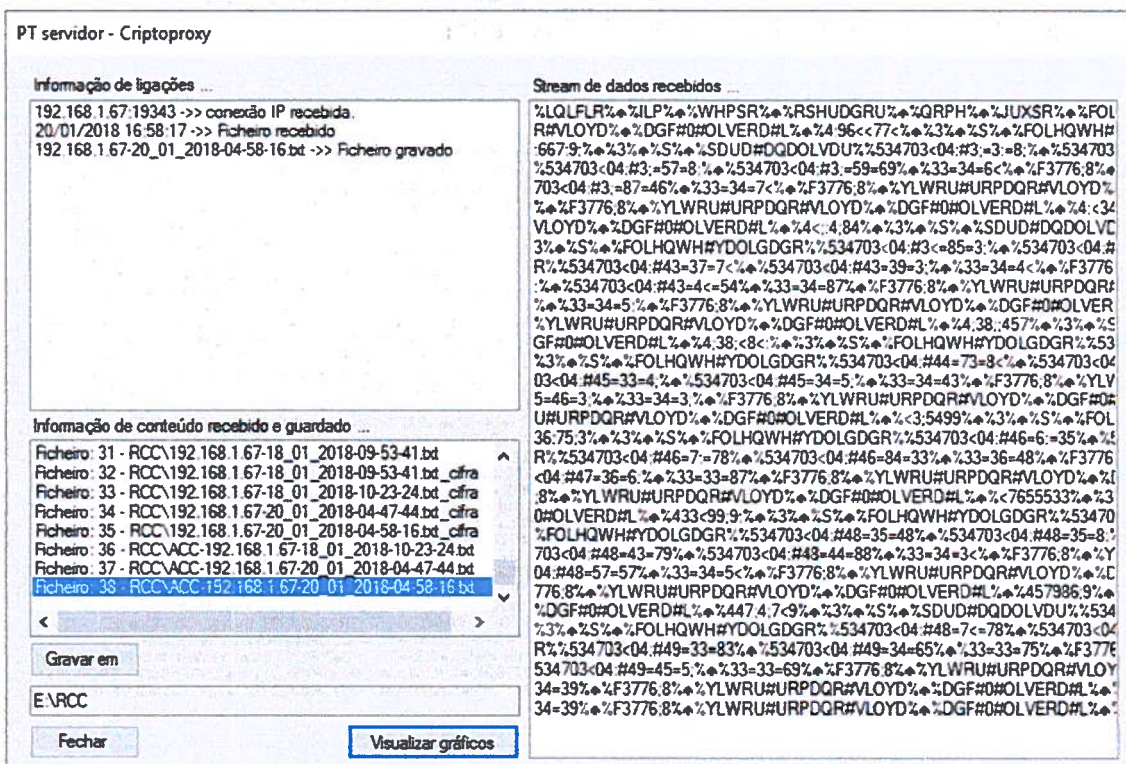


Figura 19 - visualizar informação recebida

Por último será realizado o manuseamento, tratamento e visualização dos dados.

Por compatibilidade dos sistemas de informação, o browser a utilizar é o Internet Explorer. O mesmo será executado pela aplicação “PT servidor”.

Os primeiros dados apresentados são os dados estatísticos.

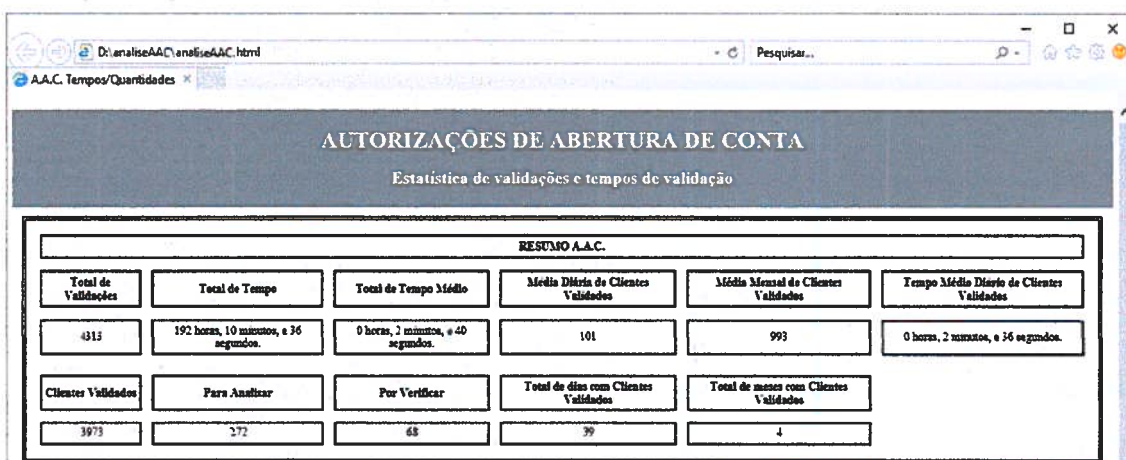


Figura 20 - análise AAC estatística

Logo de seguida, será apresentado os dados em modo gráfico.

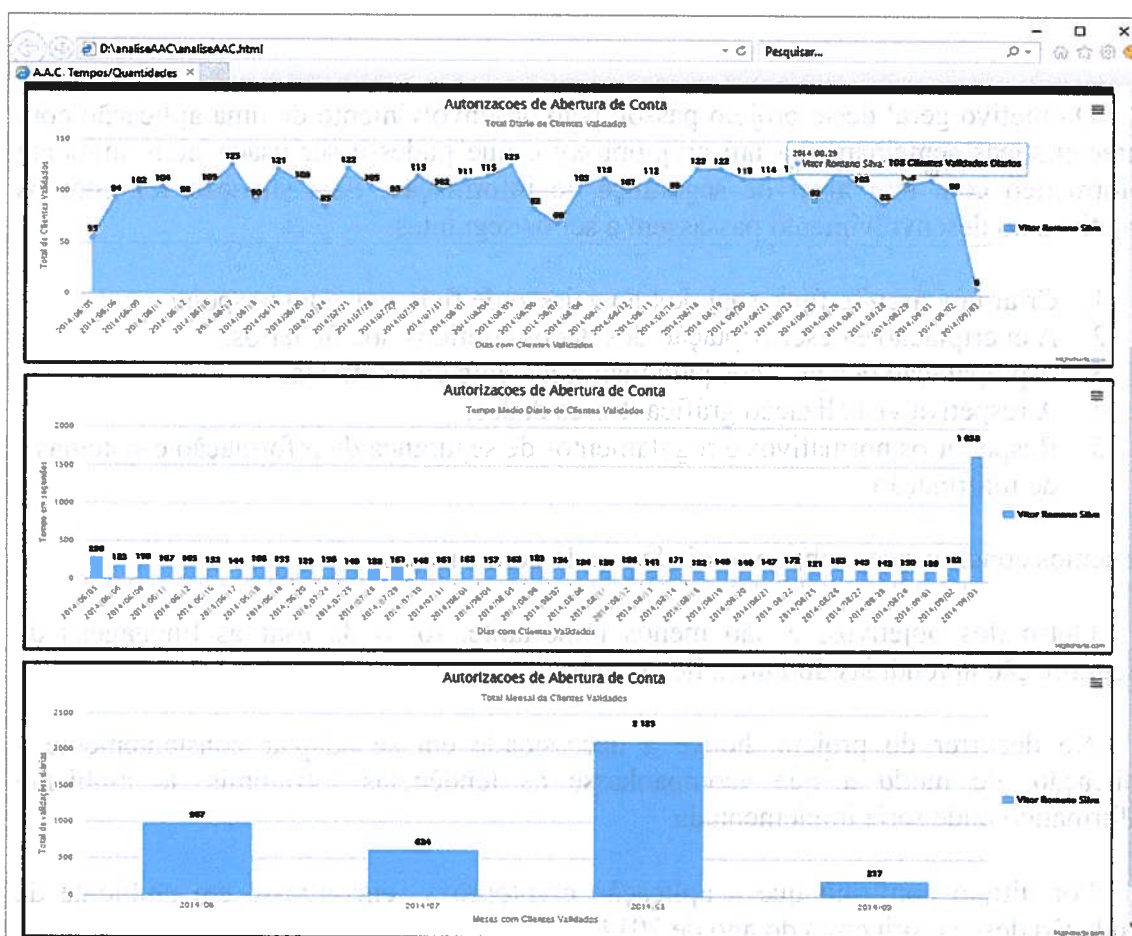


Figura 21 - análise AAC gráfica

Caso a aplicação “PT servidor” esteja ainda em execução e continue a receber mais ficheiros de dados, o utilizador só terá de voltar a seleccionar o novo ficheiro de dados na janela de visualização, e clicar novamente em Visualizar gráficos.

Com este último passo, dou por terminado a explicação do funcionamento das aplicações “PT cliente”, “PT servidor” e visualizador gráfico.

## Conclusão

O objetivo geral deste projeto passou pelo desenvolvimento de uma aplicação com características semelhantes a um cryptoproxy e que pudesse ser usada num ambiente informático com alto nível da segurança de informação. Esta situação fez com os objetivos no desenvolvimento passassem a ser os seguintes:

1. Criar um circuito funcional de transmissão de ficheiros ponto-a-ponto,
2. A encriptação e desencriptação dos dados contidos nos ficheiros,
3. A preparação dos mesmos para manuseamento em matrizes,
4. A respetiva visualização gráfica desses dados,
5. Respeitar os normativos e regulamentos de segurança da informação e sistemas de informação.

Podemos concluir que o objetivo foi alcançado com sucesso.

Outro dos objetivos, e não menos importante, foi o de usar as linguagens de programação aprendidas durante a licenciatura.

No decorrer do projeto, houve a necessidade em se adaptar constantemente a aplicação, de modo a que acompanhasse as tendências estruturais do ambiente informático onde seria implementada.

Por último confirmo que a aplicação cryptoproxy, encontra-se em ambiente de produção desde o princípio do ano de 2014.



## Webgrafia

HTML - <https://www.infoescola.com/informatica/html/>

JavaScript - <https://www.javascript.com/learn/javascript/strings>

CSS - <https://developer.mozilla.org/en-US/docs/Web/CSS>

Caesar - <http://www.cs.trincoll.edu/~crypto/historical/caesar.html>

DES - <http://searchsecurity.techtarget.com/definition/Data-Encryption-Standard>

História da Criptografia - <https://cryptoid.com.br/banco-de-noticias/a-historia-da-criptografia/>

Playfair - <https://siriarah.wordpress.com/2016/05/06/criptografia-cifra-playfair-em-python/>

Vigenère/Caesar/Scytale/Enigma - <https://cryptoid.com.br/banco-de-noticias/a-historia-da-criptografia/>

AES - <http://www.jscape.com/blog/aes-encryption>

CryptoAPI - <https://technet.microsoft.com/en-us/library/cc962093.aspx>

Top Ten Languages - <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>



The following information is provided for your information only. It is not intended to be used as a substitute for professional advice. The information is provided as a general guide only and should not be relied upon for any specific purpose. The information is provided as a general guide only and should not be relied upon for any specific purpose.

The information is provided as a general guide only and should not be relied upon for any specific purpose. The information is provided as a general guide only and should not be relied upon for any specific purpose.

The information is provided as a general guide only and should not be relied upon for any specific purpose. The information is provided as a general guide only and should not be relied upon for any specific purpose.

The information is provided as a general guide only and should not be relied upon for any specific purpose. The information is provided as a general guide only and should not be relied upon for any specific purpose.

The information is provided as a general guide only and should not be relied upon for any specific purpose. The information is provided as a general guide only and should not be relied upon for any specific purpose.