

ASC: Accettazione Smistamento e Controllo



Progetto ASC: Accettazione Smistamento e Controllo

Titolo: A.S.C. (Accettazione Smistamento e Controllo)

Autori: Rahmani Rambod, Via Pietro Casu 16, Sardegna – Sassari 07100

In collaborazione con: C'1^ SSC/RT Tampano Raffaele

Docente di riferimento: Professoressa Antonella Pazzola – Docente di Chimica.

Scuola: Liceo Scientifico “Giovanni Spano” – Via Montegrappa 2, Sassari – Tel. 079/217517

Indice dei Contenuti

- 1) Introduzione
 1. Come è nato il progetto A.S.C..
 2. Gli obbiettivi del progetto A.S.C..
- 2) Raccolta Dati
 1. La collaborazione con Raffaele Tampano.
 2. Studio della terminologia della Marina Militare.
 3. A.S.C., Un progetto costantemente verificato.
- 3) Progetto A.S.C. (Accettazione Smistamento e Controllo)
 1. Introduzione – Caratteristiche tecniche dei Softwares.
 2. Il software Asc_Exchange.
 3. Il software Asc_Server.
 4. Il software Asc_Client.
- 4) Conclusioni
 1. Possibili applicazioni del progetto.
 2. Attuale utilizzo del progetto.
- 5) Ringraziamenti

- Introduzione -

Come è nato il progetto A.S.C..

La nascita di una applicazione, di qualunque tipo essa sia, è un “evento” complesso costituito da un numero elevato di fasi fortemente legate tra loro, dove col termine “nascita” vogliamo indicare tutto l’orizzonte temporale che inizia con l’idea o la proposta di creazione di un prodotto software e termina con il suo pieno utilizzo. La realizzazione di un software passa dunque per una serie di fasi di lavoro: approfondito studio del “campo” di attinenza, raccolta dei dati di interesse, realizzazione di un algoritmo (una serie di passi che se seguiti conducono alla soluzione del problema) per ciascuna funzionalità desiderata, codifica mediante un linguaggio di programmazione dell’algoritmo in modo che venga tradotto in un programma eseguibile da un elaboratore.

L’inizio dei lavori per il progetto A.S.C. risale al Novembre 2010 quando, sotto il suggerimento del suo Capo Componente Tlc, il C’1^ SSC/RT Tampano Raffaele mi contattò parlandomi di voler fornire alla loro Unità uno specifico programma A.S.C.: creare un software per la trattazione automatica della messaggistica in arrivo ed in partenza, in quanto fino al quel momento si utilizzava un programma ormai obsoleto e non più di utilità “COMP_127”. Molto ambizioso per un ragazzo di appena 17 anni con alcuni anni di studio da autodidatta alle spalle nel campo della programmazione informatica. Realizzare un software per la Marina Militare Italiana: quale occasione migliore per mettere alla prova le conoscenze che per anni avevo studiato sui miei libri. Fortuna? Sì, se per fortuna si intende l’incontro fra la mia preparazione nel campo e l’opportunità fornitami dal C’1^ SSC/RT Tampano Raffaele. Ho deciso di provarci, ho deciso quindi di mettere a disposizione di Raffaele Tampano le mie conoscenze informatiche (senza le quali non si sarebbero avuti risultati così sbalorditivi sulla funzionalità di tale Software) e portare il progetto più in alto possibile: il progetto ASC (oggi un insieme di tre softwares) era nato.

Prima di procedere alla creazione del software vero e proprio però, come in qualsiasi progetto che si rispetti, ho dovuto studiare una buona dose di teoria. Il C’1^SSC/RT Tampano Raffaele mi ha insegnato i fondamentali della messaggistica della Marina Militare (lo standard di compilazione e distribuzione della messaggistica, la distribuzione in Gruppi/Uffici/Utenti, l’archiviazione, la gestione delle code ecc...), ho inoltre approfondito gli studi inerenti l’architettura Server/Client: la sicurezza inerente il monitoraggio dei messaggi e del loro contenuto durante l’intero trasferimento in rete era essenziale. Sicurezza nata con il progetto stesso, e migliorata sempre più durante lo sviluppo del software. Riuscire ad orientarsi in questo nuovo insieme di terminologie non è affatto stato facile, cercare di imparare un “nuovo modo di parlare” essenziale per poter portare avanti il progetto soprattutto durante le centinaia (se non migliaia) di prove cui i softwares saranno sottoposti nel periodo di progettazione. Sapevamo bene che nulla poteva essere lasciato al caso, che ogni minima modifica doveva essere testata su più computers spingendo i softwares all’estremo per poter così prevenire ogni margine di errore.

Il tutto, sia da parte mia che della mia “guida” in questo nuovo mondo, è stato colto come una sfida. Oltre un anno di lavoro in cui ho provveduto a “tradurre” le nuove conoscenze teoriche apprese in algoritmi informatici che hanno piano piano popolato le circa 15500 righe di solo codice che compongono ad oggi i tre software (Asc_Server, Client ed Exchange) del progetto A.S.C.. All’inizio era un semplice programmino di invio email con archiviazione manuale. Giorno per giorno sono stati aggiunti nuovi controlli e centinaia di righe di codice che, ad oggi, permettono ai tre softwares Asc_Server, Asc_Client e Asc_Exchange di lavorare in ricezione e in invio in completa autonomia, anche in assenza di un operatore. Oggi, se ci ripenso, è impressionante come riesca a muovermi con padronanza tra le decine di migliaia di righe di codice che per oltre un anno ho scritto, modificato e perfezionato. Il progetto A.S.C. è entrato a far parte della mia quotidianità, come una nuova “materia scolastica” da coltivare tutti i giorni tanto a livello teorico quanto a quello pratico. Una nuova materia scolastica terminata appena una settimana fa, nel febbraio 2012, che mi ha insegnato non solo nozioni teoretiche legate al settore della Marina Militare ma anche organizzative, logiche, matematiche e informatiche.

Gli obbiettivi del progetto A.S.C..

Il progetto A.S.C. è un insieme di tre softwares, (effettivamente, in questo caso, possiamo parlare di due software in quanto, come capirete più avanti, il terzo software - Asc_Exchange - ha il compito di trasferire la messaggistica tra Server e Clients), progettati affinché vi sia uno standard di distribuzione e compilazione della messaggistica da parte di tutte le Unità Navali della Marina Militare e più nello specifico dagli Operatori A.S.C.. I Softwares lavorano su uno standard di riferimento che è delineato da una pubblicazione a livello Nato "ACP 126". Gli obbiettivi del progetto, oltre quello appena citato che ritengo essere il più importante e quello per cui ho maggiormente lavorato, sono tanti e riguardano diversi ambiti, i principali sono:

1. Fornire uno standard di distribuzione e compilazione della messaggistica da parte di tutte le Unità Navali della Marina Militare secondo quanto delineato da una pubblicazione a livello Nato "ACP 126".
2. Sostituire l'ormai obsoleto programma "COMP_127": mantenersi aggiornati è fondamentale, abbiamo pensato che, gli stessi computer, sarebbe stati più funzionali con softwares aggiornati con le tecnologie più recenti possibili. Tre softwares con algoritmi il più "puliti" possibili, evitando in qualsiasi caso lo "spaghetti code".
3. Affiancare i riferimenti di funzionamento dell'applicativo Outlook Exchange e l'architettura Server/Client in modo da poter distribuire tutta la messaggistica a Gruppi/Uffici/Utenti con un unico software (Asc_Server). Il che ha diversi vantaggi, tra cui:
 - a. L'utente si abitua e impara l'utilizzo di un unico software.
 - b. Gli operatori non devono cambiare da un software all'altro per trattare i diversi tipi di messaggi perdendo ulteriore tempo.
 - c. Sicurezza dati: utilizzando un unico software, di cui si conosce l'intera struttura (sia a livello di codice che a livello funzionale), si limita al massimo un'eventuale perdita di dati.
4. La sicurezza e il monitoraggio delle attività svolte sulla postazione: la sicurezza è uno dei punti più importanti del progetto, tutti i dati vengono salvaguardati costantemente in modo da non perdere nulla anche in condizioni estreme come black out o improvviso crash dei softwares.
5. Universalità: ogni Unità Navale o Comando può utilizzare il software con la denominazione della proprio Nominativo; i tre softwares sono programmati in modo da poter essere utilizzati in tutta la Marina Militare Italiana, il che ne favorisce la diffusione in una applicazione su larga scala.
6. Totale controllo dell'operatore sul software: i softwares del progetto A.S.C. sono stati studiati e sviluppati in modo da non porre limite all'utente, ma al contrario dando all'operatore la possibilità di gestire le impostazioni e cambiarle come meglio crede:
 - a. Impostazioni di Stampa: l'utente può formattare il foglio di stampa del testo dei messaggi a suo piacere.
 - b. Modifica della Numerazione Progressiva: dal momento che tutti i messaggi trattati durante la giornata lavorativa sono numerati in Arrivo, Partenza e Intelligence, l'utente può visualizzare e modificare questa progressività.
 - c. Modifica Data: l'utente può modificare la data lavorativa per poter trattare messaggi dei giorni precedenti.
 - d. Impostazione Modalità Aggiornamento Data: l'utente decide a che ora del giorno debba essere aggiornata la data, oppure può utilizzare l'aggiornamento manuale che sincronizza la data del software con quella corrente.
 - e. Impostazioni Utenza: ogni postazione ha un nome utente e password che possono essere modificati. Si può intervenire anche sul nominativo dell'Unità Navale.

- f. Impostazioni Exchange: l'operatore, con una certa premura, può anche intervenire sulle impostazioni riguardanti il trasferimento dati tra Server e Client.

Le possibilità di controllo dell'utente sono davvero tantissime, la lista ne contiene un numero limitato.

- 7. Semplificare il lavoro degli operatori: i controlli dei tre softwares sono molto intuitivi, e soprattutto semplificano al massimo il lavoro dell'operatore:
 - a. Controllo ortografico dei messaggi.
 - b. Form di ricerca per tutti i messaggi trattati in qualsiasi data.
 - c. Tre database che contengono i messaggi trattati con possibilità di variare la data; forniscono una visione molto chiara dell'insieme dei messaggi.

Gli obbiettivi del progetto A.S.C., durante tutta la fase di sviluppo dei softwares sono stati tanti, differenti e ho affrontato tutti gli obbiettivi mettendo sempre da parte la minima possibilità che un controllo o una determinata funzione fosse impossibile da realizzare: ogni obbiettivo è stato gestito come un problema con una soluzione tanto semplice quanto funzionale e precisa.

Ad oggi sono fiero di poter affermare che non solo sono stati tutti gli obbiettivi di partenza del progetto, ma anche che durante l'oltre un anno di lavoro non ci siamo fermati davanti a niente. E molto probabilmente è questo che rende unico il progetto A.S.C. ai miei occhi: ho inserito in quei tre softwares tutte le funzionalità che avrei voluto avessero, il che, sempre e solo per me, lo rende il progetto perfetto. Il progetto A.S.C. è stato sviluppato esattamente come era stato pensato, nessuna rinuncia. Tutto ciò è stato ovviamente molto faticoso, ogni singolo pezzo dei tre softwares è stato "conquistato" poco a poco.

- Raccolta Dati -

Prima di procedere e parlare delle caratteristiche vere e proprie dei tre softwares del progetto A.S.C., dobbiamo soffermarci su una fase essenziale del progetto: la raccolta dei dati di interesse, indispensabili per poter sviluppare il progetto. Questa raccolta dati si compone di diversi punti:

- 1. La collaborazione con Raffaele Tampano: per me un vero e proprio maestro non solo durante lo studio prima della creazione dei softwares, ma che mi ha anche guidato nell'intera fase di sviluppo dei tre softwares.
- 2. Studio della terminologia della marina militare: un processo molto lungo nel quale ho appreso nozioni sulla messaggistica della Marina Militare a partire dalle basi ai concetti più difficili.
- 3. La costante verifica delle modifiche apportate ai softwares: utile non solo per non lasciare nulla al caso e spingere i softwares a condizioni estreme, ma anche perché dalle oltre centinaia di prove fatte abbiamo ottenuto dati molto utili riguardanti possibili "reazioni" dei softwares.

La collaborazione con Tampano Raffaele.

Il lavoro svolto dal C'1^ SSC/RT Tampano Raffaele è stato essenziale. Mi ha guidato in tutto il processo di studio della terminologia della Marina Militare (trattato nella sezione successiva). Vorrei sottolineare che il suo contributo di guida è stato importante tanto quanto lo sono state le mie conoscenze informatiche: senza i suoi insegnamenti infatti, non avrei mai potuto pensare di realizzare i tre softwares:

1. Come apprendere le nozioni riguardanti la messaggistica per poi trasformarli in algoritmi informatici? Uno studio da autodidatta, per non dire impossibile, sarebbe stato assai complicato.
2. Le varie sezioni di prove, sia quelle svolte da me e lui che quelle svolte presso le unità della Nave San Giorgio, sono state seguite da lui. Tutto questo ha permesso un collegamento in tempo reale mentre le prove erano in corso. Tutto questo ha permesso l'ottenimento di dati nuovi essenziali per la risoluzione dei problemi presenti.
3. Prima applicazione del progetto: facendo parte delle unità della Nave San Giorgio ha avuto la possibilità di installare il software nei suoi uffici, il che ci ha fornito l'occasione di assistere a una vera e propria applicazione del progetto. Utilizzando i tre software 24h su 24h sono emerse problematiche che con un utilizzo di poche ore non si sarebbero mai notate.

Raffaele Tampano ha cercato di semplificare il più possibile tutti i concetti (spesso ricorrendo a rappresentazioni grafiche) appositamente per me, perché la comprensione e lo studio di queste fossero più facili. Ha avuto per l'oltre un anno di lavoro che ci ha tenuto in contatto la figura di un professore, un nuovo professore che a poco a poco ha assunta la figura di un collaboratore.

Grazie alle sue altissime conoscenze nel campo, è stato in grado di istruirmi su qualsiasi punto ci fosse, spesso anche approfondendo gli argomenti in modo accurato e non lasciando nulla al caso, non dando per scontato nessuna delle conoscenze di cui avrei potuto magari avere bisogno.

Sin dai primi giorni abbiamo iniziato un "rapporto" molto professionale, non era come andare a scuola per apprendere nuove conoscenze teoriche, ed è stato proprio questo l'aspetto che credo sia stato molto fruttuoso per la formazione che questo progetto mi ha fornito: non bisogna considerare solo il risultato finale di un progetto quando questo è durato oltre un anno, ma anche tutto ciò che ci è stato "dato" da esso in questo periodo. Il progetto A.S.C. non ha infatti solo formati da punto di vista teorico ma anche insegnato un nuovo modo di collaborare, non il solito, quello presente tra i banchi di scuola con compagni e professori, ma uno più serio, professionale e soprattutto altamente fruttuoso.

Studio della terminologia della Marina Militare.

Come dall'izio puntualizzato, il progetto A.S.C. è nato diviso già in partenza su due strade: la prima, quella teorica senza la quale la seconda non potrebbe esistere, e la seconda che consiste nella realizzazione vera e propria dei software che compongono il progetto. La parte teorica principalmente consisteva nello studio della terminologia della Marina Militare. Uno studio, posso dire, molto lungo e dettagliato che ho svolto prima di iniziare a progettare. Queste nuove conoscenze che appresi contenevano, ad esempio, le diverse parti del corpo dei messaggi trattati dalle unità della Marina Militare, le diverse funzioni (l'unisci sezioni di uno stesso messaggio, diversi tipi di stampa, il gestore uffici e contatti) di cui queste unità avrebbero potuto necessitare. Lo studio, benché molto lungo e quindi faticoso, ha prodotto i suoi frutti (a mio avviso chiaramente visibili nei tre software che compongono il progetto A.S.C.). Prima di procedere e trattare un poco gli argomenti studiati, ritengo opportuno sottolineare che spesso C'1^ SSC/RT Tampano Raffaele ha semplificato al massimo termini e concetti che altrimenti sarebbero stati molto più difficili da apprendere. Per quanto riguarda gli studi svolti prima di iniziare a programmare e i tre software credo che si debba ricordare:

1. Messaggi Con Diverse Sezioni: uno degli algoritmi più difficili dell'intero progetto credo sia quello scritto per poter realizzare il cosiddetto "Unisci Sezioni", dal momento che alcuni messaggi che vengono importanti nel software possono essere divisi in diversi file (quindi parte1, parte2, parte3, ecc...) ho studiato i diversi punti da cui riconoscere le diverse parti di

- uno stesso messaggio e scritto un algoritmo che unisce le sezioni senza ripetere (tagliando le parti in determinati punti che individua) le parti già presenti.
2. **Diverse Stampe:** quando i messaggi vengono rattati, spesso è richiesto che ci sia la stampa normale o singola. Essendo diversi i messaggi ho studiato diverse tipi di stampe (testo oscurato, parzialmente oscurato, ecc...) con diverse impostazioni di layout, a seconda del messaggio appena trattato.
 3. **Numerazione dei Messaggi:** la numerazione dei messaggi era fondamentale. Tutti i messaggi che vengono trattati durante la giornata lavorata vengono classificati e ognuno ha un numero progressivo che lo identifica che va da 0000 a 9999.
 4. **Tipi di Distribuzione:** la messaggistica trattata non tutta viene inviata direttamente ai destinatari, alcuni messaggi vengono trattenuti nel computer in cui vengono scritti e l'utente riceve solo un messaggio che lo avverte di un messaggio in giacenza presso un operatore.
 5. **Uffici:** il software Asc_Server ha la possibilità di aggiunta degli uffici (in modo che gli indirizzi email o i nomi computer dei client in linea non debbano essere segnati a parte altrove). Ogni ufficio può contenere diversi indirizzi (sia indirizzi email che computer client), quindi ho studiato con l'intera rete di destinatari fosse suddivisa in contatti email, contatti client e uffici.
 6. **Databases:** il software Asc_Server contiene ben tre database che devono contenere i diversi tipi di messaggi contenuti e i messaggi vengono classificati a seconda delle loro caratteristiche.
 7. **I Messaggi:** i messaggi trattati sono differenti (Arrivi/Partenze/Intelligence/Privato) e vengono classificati a seconda delle loro caratteristiche. Per classificare i messaggi quindi ho provveduto a studiare nei minimi particolari (a volta anche riga per riga) i diversi tipi di messaggi che gli operatori potrebbero trattare.
 8. **Il Corpo dei Messaggi:** i diversi tipi di messaggi sono stati spesso da me studiati riga per riga, ad esempio ho approfondito concetti quali:
 - a. **L'FM:** la seconda riga del messaggio (composta da più parti che un algoritmo divide e esamina singolarmente), che rappresenta il mittente del messaggio. È stato predisposto anche una funzione che riconosce eventuali errori di battitura presenti nel nome del mittente.
 - b. **Il GDO:** letteralmente è il "Gruppo Data Orario", rappresenta il momento preciso in cui il messaggio viene trattato. Un algoritmo appositamente scritto lo divide in parti e le esamina singolarmente verificandone la correttezza.
 - c. **Il SIC:** rappresenta l'argomento del messaggio (oggetto espresso tramite sigla), bisogna sottolineare che i messaggi trattati dalla Marina Militare sono strutturati in maniera diversa da semplici messaggi in cui è presente un campo apposito per destinatario, oggetto, mittente ecc... Quindi sono ricorso ad un algoritmo che individua il SIC di ciascun messaggio.
 - d. **L'MSGID:** rappresenta l'argomento del messaggio (oggetto espresso tramite testo esplicito).
 - e. **Il BT:** questa funzione è essenziale, serve a dividere la componente indirizzo dalla componente testo del messaggio.
 - f. **Data Distruzione:** possono essere trattati con classifiche di segretezza elevate, e quindi necessitano una data di distribuzione, il software rileva questa riga e controlla la data di distruzione.
 - g. **Il PDC o POC:** "Punto di Contatto" o "Point of Contact", è seguito dal numero di telefono del contatto.
 - h. **Il 22-TER:** è utilizzato durante la stampa e rappresenta la lista dei messaggi trattati quel giorno nel caso si necessitasse una loro stampa.
 - i. **L'EXER:** la presenza di questa parte nel messaggio segnala la presenza di argomenti inerenti esercitazioni militari.

Queste sono solo alcune delle parti del corpo dei messaggi, per ciascuno di questi e degli altri (non citati solamente per una questione di spazio, in modo da poter trattare argomenti che ritengo di maggiore importanza) è stato scritto un apposito algoritmo che, se necessario, li divide in parti (dal momento che molti sono composti da più parti) e ne verifica l'effettiva correttezza: il messaggio verrà inviato solamente quando non saranno presenti errori.

9. Studio della Connessione Server/Client: il cuore del progetto A.S.C., ciò che lo rende di gran lunga diverso da tutti gli altri, è composto da una fusione tra l'architettura Server/Client e i riferimenti di funzionamento dell'applicativo Outlook Exchange. Prima di realizzare la vastissima architettura Server/Client presenti nei tre softwares del progetto ne ho studiate e provate diverse, i punti di riferimento che ho sempre tenuto presente sono stato:
 - a. Velocità: l'invio del maggior numero di messaggi nel minore tempo possibile.
 - b. La Sicurezza: la velocità di invio è sì importante, ma sarebbe inutile se ci fosse il rischio che i messaggi si perdessero nella rete o arrivassero alterati. Quindi ho realizzato una connessione Server/Client che tutelasse i messaggi durante l'intero loro trasferimento in rete.
 - c. Multifunzionalità: è stata realizzata un'architettura multifunzionale, e cioè che permettesse non solo di inviare e ricevere messaggi ma anche, ad esempio, controllare i clients in linea degli uffici, trasferire file tra i computer, controllare le code presenti presso Asc_Exchange, gestire le impostazioni di Asc_Exchange (che come vedrete funzione come una sorta di Server che trasferisce i messaggi tra Asc_Server e i vari Asc_Client) direttamente da Asc_Server.
10. Datazione: il progetto A.S.C., più precisamente i tre softwares, non utilizzano una sola datazione, ma bensì due tipi di datazione, una giuliana e una normale, la relazione tra queste due è stata studiata nei minimi dettagli per prevenire eventuali confusioni.

A.S.C., Un progetto costantemente verificato.

Uno dei fattori fondamentali del progetto A.S.C., che probabilmente lo ha portato a essere tanto funzionale e incredibilmente affidabile, sono state le oltre centinaia (sicuramente migliaia) di prove eseguite. Vi ho accennato già del fatto che nulla poteva essere lasciato al caso. I softwares su cui ho lavorato era sicuro sarebbe stati utilizzati (una mia speranza per il futuro di questo progetto) in luoghi (gli uffici e le unità della Marina Militare Italiana) e per trattare materiale "delicato". Quindi tutte le singole modifiche, anche quello che non riguardavano la messaggistica direttamente, sono state testate più volte su più computers. Durante questi tests, abbiamo spinto i softwares all'estremo, condizioni di utilizzo che probabilmente un operatore A.S.C. non potrà mai raggiungere, ma che ho voluto provare per vedere se i codici scritti fossero funzionali indipendentemente dal peso del traffico trattato. Non importa il numero dei messaggi trattati contemporaneamente, non importa le funzioni utilizzate o gli uffici presenti online assieme, i tre softwares reggono il traffico indifferentemente da tutti questi fattori, e inoltre non diminuiscono le prestazioni. Il raggiungimento di questo obiettivo è stato molto complicato, per molto controlli realizzati infatti notavo che non andavano bene se aumentava il numero di operatori connessi oppure con l'aumentare dei messaggi trattati o dal tipo di messaggi trattati. Ho quindi dovuto modificare, nella maggior parte dei casi riscrivere completamente, le funzioni difettanti in modo che non avessero limiti. Tuttavia, queste prove non sono state utili solamente per verificare la stabilità dei tre softwares in qualsiasi condizione e su qualsiasi computer, hanno soprattutto contribuito fornendoci una serie di dati essenziali: durante le varie prove (anche quelle svolte presso gli uffici della Nave San Giorgio, attraverso l'aiuto di Raffaele Tampano) è stato segnato tutto ciò che accadeva, e soprattutto monitorato il codice sorgente per trovare eventuali errori. L'ottimo lavoro svolto è evidente se si considera che, benché i softwaree contengano un'ottima struttura per la segnalazione degli errori (anche i minori, quindi anche semplicissimi avvisi), utilizzandoli non si ricevono mai messaggi di

errore. Un altro punto delle prove svolte da sottolineare è rappresentato dal fatto che i tre softwares sono stati testati ininterrottamente per 24h, questo grazie alla collaborazione di Raffaele Tampano che ha provveduto a installare il software sui computer della Nave San Giorgio dove sono stati utilizzati dagli operatori. Ci siamo resi conto, a volte anche con mio stupore, che alcuni errori e malfunzionamenti sono emersi solo dopo un utilizzo ininterrotto dei softwares. Le prove del progetto A.S.C. sono state spesso molto stancanti, alcune volte più stancanti anche della stessa fase di sviluppo, queste prove infatti, dal momento che bisognava testare tre software insieme, richiedevano l'utilizzo di più di un computer (spesso almeno tre o quattro) e dovevano essere ripetute più volte su più computer. Tuttavia, ho raggiunto quasi un'ossessione nell'eseguirle periodicamente e soprattutto appena veniva aggiunto una minima funzione. Non trascurandone nessuna ma costantemente occupato ad annotare i diversi "comportamenti" e risultati ottenuti. Attraverso una metodologia tanto precisa, mi sono accorto più avanti, è stato molto più facile risolvere i problemi: si individuava subito l'errore, e altrettanto velocemente si trovava una soluzione grazie l'immensa quantità di dati a disposizione.

- Progetto A.S.C. (Accettazione, Smistamento e Controllo) -

Introduzione – Caratteristiche tecniche dei Softwares.

Prima di passare a studiare i vari controlli e le caratteristiche principali dei tre softwares, trattiamo le caratteristiche tecniche che li accomunano.

I tre softwares Asc_Server, Asc_Exchange e Asc_Client sono stati sviluppati utilizzando il linguaggio di programmazione VB.NET e possono essere utilizzati solamente su computers con OS Windows (l'idea tuttavia di creare una versione utilizzabile su sistema Apple è già stata presa in considerazione). Il compilatore utilizzato è ovviamente Visual Studio (2010) e Net Framework 3.5. si consiglia tuttavia di utilizzare computer con sistema operativo Windows XP o superiori, almeno 100Mb di spazio libero sul disco su cui verrà installato il software, e almeno 256Mb di RAM.

L'architettura Server/Client, che fa affidamento al protocollo di rete TCP, è ciò che unisce i tre softwares, più precisamente ecco il compito di ciascun software:

1. Asc_Server: utilizzato per gestire i messaggi sia in uscita (dal momento che è provvisto dei riferimenti di funzionamento dell'applicativo Outlook Exchange) che in arrivo, è il cuore del progetto. Può inviare e ricevere messaggi dai vari clients in linea ed è possibile intervenire sulle impostazioni di trasferimento dati di Asc_Exchange direttamente da questa postazione.
2. Asc_Client: utilizzato per gestire (leggere, modificare, archiviare ecc...) i messaggi che riceve da Asc_Server per poi inoltrarli di nuovo, e qui Asc_Server provvederà a inviarli ai destinatari.
3. Asc_Exchange: lavora in completa autonomia (si consiglia anche l'installazione di questo software su una macchina server in quanto è possibile gestirlo interamente attraverso la postazione Asc_Server) e serve a trasferire i messaggi da Asc_Server ai vari Asc_Client.

Come potete intuire quindi, viene utilizzata l'architettura Server/Client per permettere una "collaborazione" dei vari operatori per la gestione dei messaggi, mentre i riferimenti di funzionamento dell'applicativo Outlook Exchange sono utilizzati quando il messaggio è pronto ad essere inviato ai destinatari esterni. Tutti e tre i softwares (possiamo anche parlare di due softwares – Asc_Server e Asc_Client – dal momento che Asc_Exchange non ha assolutamente bisogno di operatore) possono lavorare in completa autonomia in assenza di operatore. Ricevere i messaggi che viene loro inviato e archivarli in attesa che vengano trattati da un operatore. I

messaggi inviati vengono conservati da Asc_Exchange sino a quando il destinatario (Asc_Server o uno degli Asc_Client) sarà online e riceverà il messaggio.

Il software Asc_Exchange.

Prima di Asc_Server e Asc_Client che hanno alcune funzioni (quelle inerenti i controlli “ortografici” del corpo dei messaggi) identiche, studiamo le caratteristiche di Asc_Exchange.

Questo software, il quale sviluppo è stato terminato per ultimo in quanto i lavori l’architettura Server/Client sono stati terminati per ultimati a causa di una costante ricerca di sicurezza e allo stesso tempo di velocità, si chiama “Asc_Exchange” e serve a trasferire i messaggi tra Server e Clients. Funziona quindi come una sorta di server online (molto simile a un gestore mail quindi come Yahoo Mail o Hotmail) per lo scambio di messaggi tra utenti, disponibile 24h con la differenza che lavora su una rete locale.

Il software Asc_Exchange è composto da quattro forms:

1. Accesso: serve per accedere al software attraverso credenziali che vengono impostate al primo utilizzo del software e richiede ogni volta che viene avviato.
2. Utenza: form utilizzato per il primo ovvio del software per la scelta delle credenziali di accesso e per una eventuale modifica.
3. Impostazioni: questo form offre all’utente la possibilità di intervenire sulle impostazioni inerenti la gestione dei messaggi da parte di Asc_Exchange; possono essere modificate:
 - a. Frequenza Controllo Code: ovvero la frequenza (espressa in secondi) con la quale Asc_Exchange controllerà la presenza di messaggi da inviare.
 - b. Frequenza Invio Messaggi: ovvero la frequenza (espressa in secondi) con la quale vengono inviati i messaggi.
4. Main: è il form principale e contiene i seguenti controlli:
 - a. Label con lo stato di connessione di Asc_Exchange.
 - b. Label con l’IP della macchina (utilizzato da Asc_Server e dai vari Asc_Client per connettersi).
 - c. Label con lo stato di connessione di Asc_Server.
 - d. Listbox contenente i nomi delle macchine Client connesse.

Asc_Exchange ha inoltre la classe “clsListener.vb”, che li permette di monitorare lo stato di connettività dei vari clients in rete, e anche di poter accettare le connessioni dei clients, e la classe “tcpConnection.vb” utilizzata per il trasferimento di dati. Queste due (“clsListener.vb” e “tcpConnection.vb”) sono le classi che ho appositamente creato per l’architettura Server/Client del progetto. I processi di ricezione e invio di Asc_Exchange sono stati studiati per ricevere e inviare contemporaneamente con assoluta facilità e, anch’essi, sono stati sviluppando cercando di inviare il maggior numero di messaggi nel minor tempo, ma allo stesso tempo in totale sicurezza. I messaggi ricevuti vengono conservati nella cartella “temp” (cartella che viene creata, se non presente, all’avvio del software nella cartella di avvio). I file presenti in questa cartella sono gestiti dal processo “Smista” che ha il compito di dividerli a seconda che provengano da Asc_Server o da un Asc_Client rispettivamente nelle cartelle “Messaggi Ricevuti Dai Clients” (dalla quale verranno inoltrati ad Asc_Server) e “Messaggi Ricevuti Dal Server”. In quest’ultima cartella subiscono un’ulteriore divisione, i messaggi vengono suddivisi in cartelle a seconda del client a cui sono indirizzati. Il processo di invio messaggi (che inizia con l’invio dei messaggi in attesa per il server e procede con i messaggi presenti per ciascun client) è un ciclo loop continuo che, a intervalli impostabili, esegue il controllo e procede all’eventuale invio in caso di destinatario connesso. Asc_Exchange inoltre ha tutta una serie di funzionalità per fornire dati ad Asc_Server che tratterò quando approfondiremo quest’ultimo.

Il software Asc_Server.

Prima di procedere e trattare i vari controlli di Asc_Server, trattiamo la sua architettura di trattazione messaggi e quella Server/Client: Innanzitutto, ha a disposizione due tipi di trattazione, quella Manuale (che salva i messaggi in locali e i destinatari ricevono solamente un messaggio che li avverte della presenza di un messaggio per loro presso Asc_Server), e quella Automatica che invece invia i messaggi utilizzando o i riferimenti di funzionamento dell'applicativo Outlook Exchange oppure l'architettura Server/Client. Per quanto riguarda il processo di invio dei messaggi ai vari Asc_Client (effettivamente i messaggi vengono inviati ad Asc_Exchange che li smista e provvede a inoltrarli ai rispettivi destinatari), questi vengono in un primo momento salvati all'interno della cartella "temppartenze" di Asc_Server (cartella che si trova nella cartella di avvio del software e viene creata al suo avvio), non appena la lista dei messaggi da inoltrare ad Asc_Exchange è completa, sfruttando la classe "tcpConnection.vb", Asc_Server inizia il processo di invio con una frequenza di un messaggio ogni due secondi (questa è la frequenza di default scelta da me per non intasare il ricevimento di messaggi da parte di Asc_Exchange, dobbiamo tenere conto anche del fatto che Asc_Exchange nello stesso istante potrebbe ricevere messaggi anche dai tantissimi Asc_Client connessi), frequenza che può essere modificata a piacimento dell'utente. Oltre alla possibilità di inviare messaggi ai vari Asc_Client, Asc_Server può anche riceverli: i messaggi che vengono inoltrati ad Asc_Server da Asc_Exchange vengono in un primo momento conservati presso la cartella "temp", in un secondo momento vanno in contro al processo "Smista" e verranno spostati nella cartella "Messaggi Ricevuti" e saranno poi visibile nella Listview che si trova nel Form principale di Asc_Server. Sempre inerenti l'architettura Server/Client sono da ricordare:

1. La possibilità di controllare gli Asc_Client in linea in ogni momento e aggiungere quelli desiderati agli uffici.
2. Un controllo automatico che tiene aggiornati i messaggi in attesa per tutti gli uffici presenti e lo stato di connessione dei loro contatti.
3. Possibilità di gestire le code: da Asc_Server è possibile controllare totalmente (si possono vedere i messaggi in attesa, e anche cancellarli) i messaggi presenti presso Asc_Exchange in attesa di essere inviati.

Per quanto riguarda i controlli di Asc_Server credo che siano da ricordare:

1. Introduzione dei messaggi nei formati ACP-126. L'introduzione può avvenire tramite tastiera, disco magnetico o penna USB.
2. Distribuzione dei messaggi in Partenza/Arrivo/Intelligence ACP-126 direttamente agli uffici collegati sia tramite indirizzo di posta elettronica, quindi sfruttando Outlook Exchange, sia tramite AsClient un software creato per interagire sia in ricezione che in trasmissione con AsC_Server.
3. Archiviazione giornaliera dei messaggi in Partenza/Arrivo/Intelligence.
4. Stampa 22-ter Arrivo/Partenza.
5. Stampa 22-ter Intelligence.
6. Archiviazione Messaggistica PCN (Testo Figurativo).
7. Archiviazione Messaggistica PCN (Testo Completo).
8. Invio E-mail privata.
9. Funzione Unisci Sezioni.
10. Controllo standard Acp 126 per la messaggistica in Partenza.
11. Funzione Uffici.
12. Funzione Cerca.
13. Funzione Allegati

Il software Asc_Client.

Anche per il software Asc_Client inizierò parlando della struttura Server/Client (unico mezzo di trasferimento dei messaggi per Asc_Client) per poi parlare dei controlli che la contraddistinguono.

Asc_Client ha un invio di messaggi molto più diretto rispetto ad Asc_Server, dovendo infatti inoltrare messaggi ad un unico destinatario (Asc_Server), non c'è alcun passaggio per una cartella temporanea. Il file contenente il messaggio viene creato e inviato nello stesso momento in cui l'operatore utilizza il pulsante adibito all'invio di messaggi in Asc_Server. Anch'esso utilizza la classe "tcpConnection.vb" per i trasferimenti. Anche Asc_Client ha la possibilità di ricevere messaggi da Asc_Server: i messaggi ricevuti vengono salvati nella cartella "temp" e da cui andranno in controllo al processo "Smista". Questo processo smista i file salvandoli nella cartella "Messaggi Ricevuti" e saranno visibili nella Listview presente nel Form principale di Asc_Client.

Oltre alla sua architettura Server/Client, Asc_Client si distingue soprattutto per la costituzione del suo Form principale. Questo contiene infatti due aree (delimitate da due Groupbox), la prima "Invio in Asc" serve ad inviare i messaggi ad Asc_Server, la seconda "Area Modifica – Ricezione", serve per modificare i messaggi ricevuti da Asc_Server per poi rinviarli pronti ad essere inviati ai destinatari. Una delle funzioni totalmente originali del software Asc_Client, è la gestione delle code dei messaggi ricevuti da Asc_Server che sono in attesa di essere visionati. Questo controllo si trova nell'area "Area Modifica – Ricezione" ed è composto da un gruppo di Radiobutton (che rappresentano ciascuno una delle classifiche dell'FM, criterio secondo quale i messaggi in attesa vengono catalogati), e una Listbox che, a seconda del Radiobutton selezionato, mostra un determinato gruppo di messaggi (ovviamente quelli con l'FM con la classifica selezionata).

- Conclusioni -

Possibili applicazioni del progetto.

Nel Novembre del 2010, quando iniziai gli studi per poter poi passare allo sviluppo dei softwares del progetto A.S.C. non avrei mai potuto pensare che avrei lavorato per oltre un anno su quel progetto e che dopo appena un paio di mesi di sviluppo sarebbe stato utilizzato dalle unità della Nave San Giorgio aggiornando giorno dopo giorno la versione del software con quelle da me fornite per poter avere le ultime funzionalità inserite. La possibile applicazione del progetto A.S.C. quindi è effettivamente possibile e molto presto spero che possa diffondersi in tutta la Marina Militare Italiana. Per raggiungere questo ultimo e assai ambito obiettivo, sempre grazie alla collaborazione del C'1^ SSC/RT Tampano Raffaele, abbiamo proposto il progetto presso la sede di comando della Marina Militare a Roma. Dopo alcuni riscontri positivi aspettiamo che il comando a Roma invii la sua proposta di acquisto alle unità navali e che esse aderiscano all'utilizzo di questo software.

Attuale utilizzo del progetto.

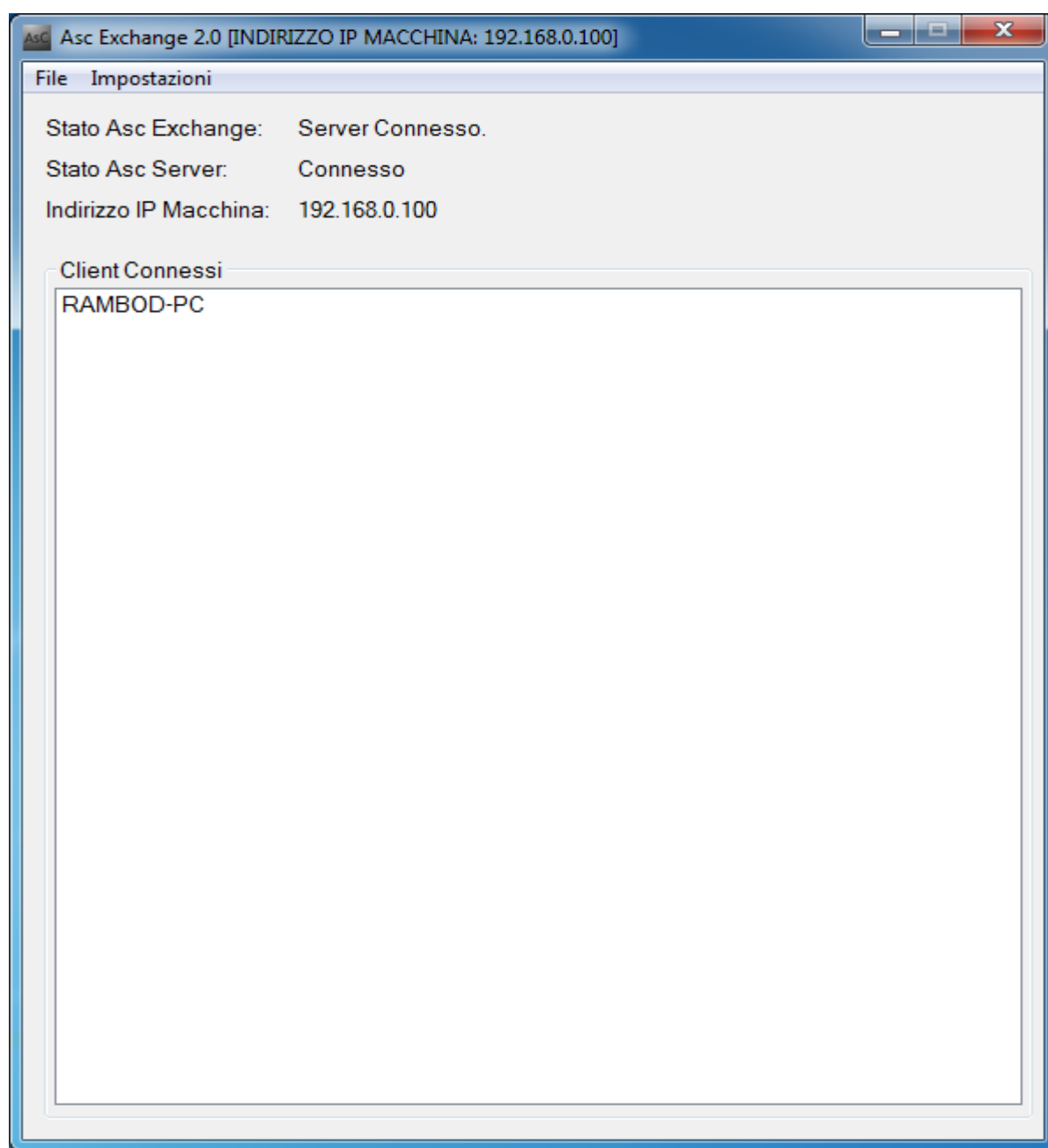
Attualmente il progetto, come vi ho già accennato precedentemente, viene utilizzato presso la Nave San Giorgio nel quale è stato adottato a partire dai primi mesi di vita dei softwares. Unità navale che, posso dirlo, ha visto l'intero sviluppo dei tre softwares sino alla versione attuale, la più stabile, che utilizzano.

- Ringraziamenti -

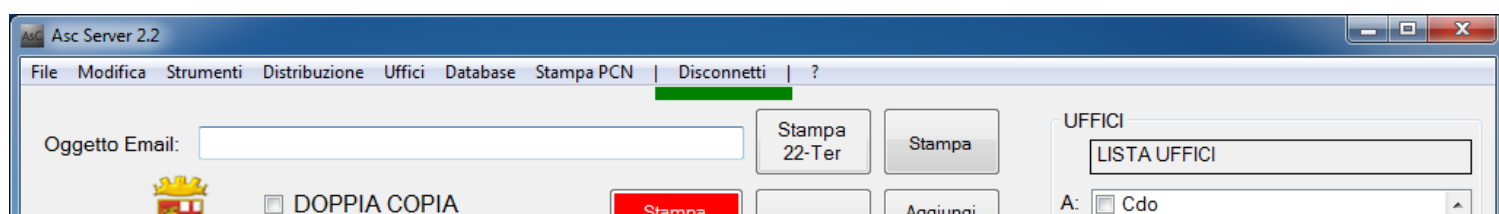
Ringrazio il C'1^ SSC/RT Tampano Raffaele per il supporto offertomi, e riconosco che senza la sua collaborazione il progetto A.S.C. non sarebbe mai stato possibile. Lo ringrazio per i tantissimi insegnamenti che mi ha dato e per la pazienza e la voglia di collaborazione dimostrata in tutto il lunghissimo periodo di sviluppo del progetto.

- Alcuni Screen dei tre Softwares -

Asc_Exchange



ASC_SERVER



Asc_Client

