

Functional and technical specifications of the TicketBAI 1.1 system

CONTENTS

1	INTRODUCTION	4
1.1	What it is TicketBAI?	4
1.2	Area of application	4
2	ABBREVIATIONS AND DEFINITIONS.....	5
3	GROUPS OF TECHNICAL SPECIFICATIONS	7
3.1	Register of TBAI invoicing software developers (Developers Register)	7
3.2	Issuing the invoice	7
3.3	Sending information to the tax authorities	8
3.4	On-site Verification.....	8
4	TECHNICAL SPECIFICATIONS OF THE INVOICE ISSUE PHASE	9
4.1	Generating the TBAI file	9
4.2	TBAI file signature.....	10
4.3	Generating the invoice image	10
4.3.1	TBAI requirements for generating the invoice image	10
4.3.2	TBAI identifier.....	12
4.3.3	TBAI QR.....	13
5	TECHNICAL SPECIFICATIONS OF THE ON-SITE VERIFICATION PHASE	16
6	OTHER REQUIREMENTS	17
6.1	General Specifications	17

6.2	Signing the TBAI invoicing software	17
6.3	Application versioning	18
6.4	Recommendations on completing the invoice series and number fields	18
6.5	Cancelling an invoice	18
7	DOCUMENT MANAGEMENT	20
7.1	Managing specifications and versioning	20
7.2	Signature policy	20
8	APPENDIX 1 – STRUCTURE OF THE TICKETBAI FILE	21
9	APPENDIX 2 – STRUCTURE OF THE TICKETBAI CANCELLATION FILE	31
10	APPENDIX 3 – ERROR-CHECKING CRC ALGORITHM.....	33

1 INTRODUCTION

1.1 What it is TicketBAI?

TicketBAI is a shared project by the Regional (Provincial) Finance Authorities and the Basque Government to implement a series of legal and technical obligations regarding taxpayers' invoicing software. It will enable the tax authorities to monitor taxpayers' business income, particularly in sectors that deliver goods or perform services to end consumers that are largely paid for in cash.

Two of the main goals of the Basque authorities are to assist taxpayers in fulfilling their fiscal obligations and to combat tax fraud, thus ensuring that everyone contributes to public spending according to their financial means. This is the essential pillar of the welfare state, the model of coexistence that enjoys the greatest social and political consensus in this country.

The different public institutions are working constantly to improve and update their systems and working procedures, to make it as easy as possible for taxpayers to fulfil their obligations. At the same time, they are devoting their best efforts to enhancing the effectiveness and efficiency of the fight against tax fraud and against individuals who do not voluntarily fulfil their tax obligations, in a clear breach of solidarity with their fellow citizens.

To achieve these goals, the Basque provincial finance departments are using all available developments information and communication technology. The TicketBAI project is one of the most innovative elements in the field of assistance and monitoring of taxpayers with businesses. It aims to ensure that all transactions conducted by individuals and organisations in business are properly declared and draws on the experience of other countries in the region and the recommendations of OECD reports "Electronic Sales Suppression: A threat to tax revenues" and "Technology Tools to Tackle Tax Evasion and Tax Fraud" published in 2013 and 2017, respectively, as well as the report on "Implementing Online Cash Registers: Benefits, Considerations and Guidance", published on 28 March, 2019.

1.2 Area of application

The TicketBAI system affects all natural and legal persons and entities without legal personality (hereinafter referred to as legal persons and entities) that conduct business activities and fall under the jurisdiction of the Basque provincial finance departments in the area of Private Income or Company Tax (under the terms of the Economic Accord with the Autonomous Community of the Basque Country, enacted by Act 12/2002 of 23 May 2002).

Once the TicketBAI system comes into force, all persons and entities that issue invoices will be required to use devices such as Point of Sale (POS) systems, personal computers, advanced vending machines, advanced weighing scales, tablets, mobile phones and other devices, which must comply with the technical regulations set out in this document.

The regulatory standards will determine which taxpayers are totally or partially exempt from compliance with TicketBAI system obligations, according to the transactions identified.

2 ABBREVIATIONS AND DEFINITIONS

For the purposes of this document, the terms below shall be defined as follows:

TBAI: abbreviation of TicketBAI.

Invoicing standards:

In the Historical Territory of Araba, the regulations governing invoicing obligations, passed into law by Decree of the Territorial Government of Araba 18/2013, of 28 May 2013.

In the Historical Territory of Bizkaia, the regulation governing invoicing obligations, passed into law by Decree of the Territorial Government of Bizkaia 4/2013, of 22 January 2013.

In the Historical Territory of Gipuzkoa, the regulation governing invoicing obligations, passed into law Decree of the Territorial Government of Bizkaia 8/2013, of 26 February 2013.

Invoice: Invoice on paper or in printable electronic form, issued in accordance with invoicing standards or any other document on paper or in printable electronic form, with the same content as the invoice, accrediting the delivery of goods or provision of services by a taxpayer subject to the TBAI system in cases where is no obligation to issue an invoice.

TBAI File: XML file obtained before an invoice is issued, containing the data defined in this document.

TBAI invoicing software: set of computer programs, instructions and rules that fulfil the functional and technical specifications established in this document, prepared by a developer and entered in the register of developers of TBAI invoicing software (the "Developers Register").

QR Code: two-dimensional bar code capable of storing encoded data.

XML (eXtensible Markup Language): a markup language derived from SGML which is used to define document structures.

Developer: natural or legal person or entity without legal personality who has developed TBAI invoicing software for distribution on the market, is entered in the register of TBAI invoicing software developers and has signed a binding declaration of compliance with the requirements of the TBAI system.

In cases where the TBAI invoicing software is developed by the taxpayers themselves, they must be entered in the register of TBAI invoicing software developers.

Taxpayer: natural person who conducts a business activity in accordance with the legislation on Private Income Tax or a legal entity or entity without legal personality that is subject to Company Tax.

Tax authority: Basque provincial authority with legislative jurisdiction over the taxpayer's Private Income Tax or Company Tax.

Invoicing device: device from which the invoicing application is accessed, whether or not it accesses a remote server for this purpose. Example: in the case of a web-based application, the serial number of the invoicing device will be that of the device on which the browser is installed, and not the remote server which it accesses to operate the application.

3 GROUPS OF TECHNICAL SPECIFICATIONS

TBAI technical specifications are divided into a number of groups. These groups correspond to different moments in the implementation of the TBAI system.

3.1 Register of TBAI invoicing software developers (Developers Register)

Any person or entity wishing to distribute TBAI invoicing software on the market must be entered in the register of TBAI invoicing software developers (Developers Register) before it can sell or distribute said software.

In cases where the TBAI invoicing software is developed by the taxpayers themselves, they must be entered in the register of TBAI invoicing software developers.

The developer must be entered in the register and sign a binding declaration for each TBAI invoicing software package it develops, stating that it meets the requirements set out in the regulatory standards for the TBAI system.

Once the software has been registered, a TBAI Licence will be issued which must be included in the business generated by the invoicing software, as specified in Section 4.1 of this document (Generating the TBAI file).

When making the entry in the register, the tax authorities will not check or certify the invoicing software.

The purpose of the Developers Register is to have a list of developers who have undertaken to distribute invoicing software that meets TBAI specifications.

Each of the three Basque provincial finance departments will have its own register of TBAI invoicing software developers. The developer and its TBAI invoicing software can be registered in any of these.

The three registers will be interconnected, so that entry in one register will entitle the developer to distribute the software among any taxpayers in the three jurisdictions.

The list of registered developers and software will be publicly available, and may be consulted on the website of the Basque tax authorities.

3.2 Issuing the invoice

All invoicing software used to implement TBAI requirements must follow the process detailed below each time an invoice is issued:

- An XML file must be generated that matches the design and content of the TBAI format schema. This file will be known as the TBAI file.
- The TBAI file must be electronically signed.

- The invoice image may be generated either physically or in electronic format. The invoice image must include part of the signature of the TBAI file.

The procedure MUST be in the following order: generation of the TBAI file; signing of the file; generation of the image on paper or in electronic format.

3.3 Sending information to the tax authorities

All signed TBAI files must be sent to the corresponding tax authority. Each tax authority will publish specific standards, setting out the technical requirements and deadlines for submitting the information. These may vary from one authority to another.

3.4 On-site Verification

The invoicing software used to implement TBAI requirements must include functions that allow the TBAI system to be verified by staff from the finance department. These functions are listed in Section 5 of this document.

4 TECHNICAL SPECIFICATIONS OF THE INVOICE ISSUE PHASE

4.1 Generating the TBAI file

During the process of issuing an invoice, and before the invoice image is generated, the invoicing software must generate an XML file matching the design and content of the TBAI format schema. UTF-8 encoding must be used.

A detailed schema of the TBAI file to be issued may be found in Appendix 1 of this document. In general terms, the blocks of information in the file are as follows:

- Subjects: issuer, recipient, several recipients, invoice issued by third parties or recipient, etc.
- Invoice – Invoice Heading: includes, among others fields: invoice series, invoice number, issue date, issue time, simplified invoice, invoice issued in substitution of simplified invoice, correcting invoice, corrected invoices, etc.
- Invoice - Invoice data: among others fields, includes: transaction date, invoice description, total invoice amount, tax base at cost, VAT regime codes and taxable transactions, etc.
- Invoice – Itemisation at invoice level: among others fields, includes: subject, exempt, grounds for exemption, exempt tax base, non-exempt rate, etc.
- TBAI fingerprints:
 - Chaining with previous invoice.
 - Series, invoice number and issue date of the previous invoice.
 - Signature of the TBAI file for the previous invoice. The SignatureValue field in the TBAI file of the previous invoice will be truncated to the first hundred characters to prevent an unnecessary increase in size.
 - TBAI Licence: provided by the TBAI developers register.
 - Developer entered in the TBAI developers register.
 - Invoicing software entered in the TBAI developers register.
 - Device.
 - Serial number of the device issuing the invoices. If the serial number is longer than the size of the field assigned in the TBAI file, the last thirty characters will be entered.

- The invoice issuing device is taken to be the device from which the invoicing application is accessed, whether or not it accesses a remote server for this purpose. For example, in the case of a web-based application, the serial number of the invoicing device will be that of the device on which the browser is installed, and not the remote server which it accesses to operate the application.
- During a transitional period, this entry will be optional. In the future, the possibility will be assessed of making it obligatory, depending on the possibility of automating the obtention of this datum on all architectures.

4.2 TBAI file signature

Each TBAI file must be electronically signed. Part of the TBAI file signature must be included in the invoice image. For this reason, the signature process always comes before the generation of the image.

The specifications for the TBAI file signature are included in the document “TicketBAI Signature Policy”. All items contained in that document are covered in the specifications for the TBAI system. Among other items, the signature policy sets out:

- The electronic certificates that can be used.
- Where these certificates should be installed and where the signature should be made, depending on the type of certificate used.

Amendments to the “TicketBAI Signature Policy” document are managed independently and may be made either in conjunction with this document or separately.

4.3 Generating the invoice image

4.3.1 TBAI requirements for generating the invoice image

The TBAI requirements on invoice image generation will be mandatory and will apply to both invoice images on paper and in electronic format (e.g. PDF).

These specifications do not amend the invoicing standards or introduce any additional requirements. The TBAI system specifications consists exclusively of obligations that must be fulfilled by the taxpayer issuing the invoice, within the area of Private Income Tax and Company Tax.

The invoicing software must generate a TBAI identifier and a TBAI QR Code, which must be included in the invoice. These items must comply with the specifications contained in this section and paragraphs 4.3.2 and 4.3.3 of this document:

- The TBAI identifier consists of a code, consisting of numbers, letters and other characters, identifying the invoice in the TBAI system. The font type and size must be similar to the rest of the invoice, to ensure that the identifier can be read by the recipient.
- The TBAI QR Code consists of a code in QR format (hereinafter referred to as the TBAI QR). It must be no larger than 30x30 millimetres and no smaller than 40x40 millimetres in size.

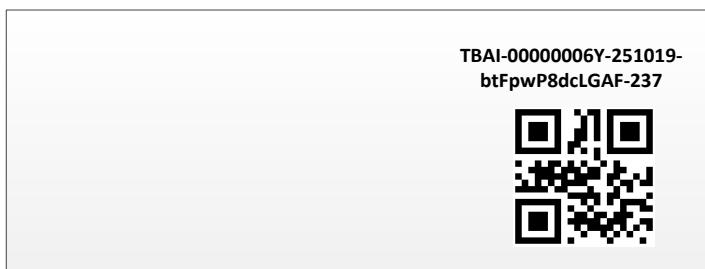
The position of the TBAI identifier and TBAI QR will depend on the orientation of the invoice:

- In portrait (vertical) orientation, they should be positioned at the bottom of the invoice. The TBAI identifier should be included in a single line, above the TBAI QR.
- In landscape (horizontal) orientation, they should be positioned on the right-hand side of the invoice. The TBAI identifier should be included in a single line, above the TBAI QR.

If it is impossible to fit the TBAI identifier on a single line, it will be permissible to use several consecutive lines. In this case, the last character on each line (except the last line), should consist of the separator “-” (hyphen).

The images below are solely intended as examples of where to position the TBAI identifier and TBAI QR in the invoice. The content, size and proportions of these examples are not valid.

Landscape (horizontal) orientation:



Portrait (vertical) orientation:



4.3.2 TBAI identifier

Identifies the invoice in the system and ensures that the invoice is associated with its corresponding TBAI file.

It has a fixed length of 39 characters, as follows:

- 4 characters of fixed text in upper-case letters: TBAI.
- 1 character “-” as a separator. Hyphen.
- 9 characters for the taxpayer's VAT Registration Number. This must match the VAT Registration Number, in its official format, contained in the TBAI file.
- 1 character “-” as a separator. Hyphen.
- 6 characters for the invoice issue date. This must match the date entered in the TBAI file (“FechaExpedicionFactura” tag). It must be in DDMMYY format, with no internal separators. Where necessary, each of the subfields should be left-padded with zeros, so that the date is always 6 numbers in length (e.g. 010121 for 1st January, 2021).
 - DD: invoice issue day.
 - MM: invoice issue month.
 - YY: Last two digits of the invoice issue year (e.g. for 2021, YY=21).

- 1 character “-” as a separator. Hyphen.
- 13 characters of the TBAI file signature. The first thirteen characters of the SignatureValue field of the TBAI file associated with the invoice.
- 1 character “-” as a separator. Hyphen.
- 3-character error-detecting code (CRC) to ensure that the identifier is correct:
 - o The CRC should be calculated by the invoicing software.
 - o It is obtained by applying the CRC-8 algorithm to the previously defined string of characters (the previous 36 characters).
 - o The algorithm input is the content of the identifier generated thus far (the first 36 characters of the identifier) with UTF-8 encoding.
 - o The algorithm output will be in decimal format, left-padding the 3 last characters of the identifier with zeros (where necessary).
 - o The algorithm used by the tax authorities to check the CRC can be found in Appendix 3 of this document. This algorithm has been made public in order to ensure that the invoicing software generates the same results as those obtained by the tax authorities.

Generic example of TBAI identifier: **TBAI-NNNNNNNNN-DDMMYY-FFFFFFFFFFFF-CRC**

Specific example of TBAI identifier: **TBAI-00000006Y-251019-btFpwP8dcLGAF-237**

(The VAT Registration Number and signature fields shown in these examples are invalid and are only intended as examples of the format to be used).

4.3.3 TBAI QR

Allows the recipient to obtain the fiscal data associated with the invoice. Taxpayers are responsible for ensuring the readability of the QR codes on invoices issued in their business. An invoice with an unreadable QR will be considered invalid in terms of TBAI system requirements.

The QR must have an error correction level of M. The code must be generated with UTF-8 encoding.

The colour contrast between the QR and the background must be high enough to ensure readability. A blank space of 6 millimetres should therefore be left around the four sides of the QR Code.

The QR must contain a valid URL for accessing the web-based TBAI invoice checking application with data from the invoice included as parameters. If the URL or its parameters contain non-valid characters, they should be correctly “coded” (using URL encoding), in accordance with standard web architecture practise.

The content of the QR Code will be as follows:

- URL of the specific QR-reading web application of the tax authority in question:
 - Araba-Álava: <https://ticketbai.araba.eus/TBAI/RTBAI> (without "/" for calculating the CRC).
 - Bizkaia: <https://batuz.eus/RTBAI/> (with final "/" for calculating the CRC).
 - Gipuzkoa: <https://tbai.egoitza.gipuzkoa.eus/qr/> (with final "/" for calculating the CRC).
- Parameters:
 - TBAI identifier.
 - Code: **id**
 - Value: The characters defined in the TBAI Identifier section.
 - Invoice series.
 - Code: **s**
 - Value: invoice series, in accordance with invoicing standards. Must match the series in the TBAI file ("SerieFactura" field).
 - Invoice number.
 - Code: **nf**
 - Value: invoice number, in accordance with invoicing standards. Must match the invoice number in the TBAI file ("NumFactura" tag).
 - Total invoice amount.
 - Code: **i**
 - Value: invoice amount, including VAT. Must match the value and format of the total amount in the TBAI file ("ImporteTotalFactura" tag).
 - CRC-8.
 - Code: **cr**
 - Value: error-detecting code (CRC), to detect accidental changes to the content of the TBAI QR. The CRC should be calculated by the invoicing software.

It will be included as the last parameter in the URL. The CRC is obtained by applying the CRC-8 algorithm to the string of characters contained in the QR.

The algorithm input will be the content of the QR generated thus far with UTF-8 encoding. It will therefore not include either the cr parameter itself or its associated "&" symbol used to add it to the other parameters (query string).

The output of the algorithm will be written in decimal format as a new parameter of the URL.

The algorithm used by the tax authorities to check the CRC can be found in Appendix 3 of this document. This algorithm has been made public in order to ensure that the invoicing software generates the same results as those obtained by the tax authorities.

Example of TBAI QR content:

<https://batuz.eus/QRBTBAI/?id=TBAI-00000006Y-251019-btFpwp8dcLGAF-237&s=T&nf=27174&i=4.70&cr=007>

Example of the TicketBAI QR:



5 TECHNICAL SPECIFICATIONS OF THE ON-SITE VERIFICATION PHASE

The TBAI invoicing software must include functions that facilitate rapid verification of TBAI specifications.

Inspectors from the tax authority will perform on-site checks and request that certain general information be obtained on the TBAI invoicing software and the invoicing device.

This information must match the information on the TBAI files generated by the device. The TBAI invoicing software must show the following information on a unique screen:

- Developer of the TBAI invoicing software used from the device. Must match the information on the TBAI developers register.
 - o VAT Registration Number
 - o Trading Name.
- Name of the TBAI invoicing software used from the device. Must match the information on the TBAI developers register.
- Version of the TBAI invoicing software used from the device.
- Serial number of the device. For a transitional period, displaying this serial number will be optional. In the future, the possibility will be assessed of making it obligatory, depending on the possibility of automating the obtention of this datum on all architectures.

During the on-site verification, inspectors may check that the invoicing software is operating correctly and may ask to be provided with the TBAI files generated, including both those created in the verification process and any that the taxpayer is required to keep in accordance with general tax regulations.

6 OTHER REQUIREMENTS

6.1 General Specifications

One device may issue invoices for several different taxpayers.

In test mode, the TBAI invoicing software must not create or sign TBAI files or generate invoice images with TBAI requirements. All files and invoice images created must correspond to actual transactions. Non-definitive invoices (for example, proforma invoices) that include the TBAI identifier and/or the TBAI QR Code will not be accepted.

All invoices issued using the TBAI invoicing software must comply with TBAI legislation and specifications.

Once generated, an invoice may not be re-issued. Should it be necessary to reissue an invoice, the new invoice must clearly identify that it is a duplicate of a previous one. The issuing of duplicate invoices is regulated by the invoicing standards, which stipulate that:

- Taxpayers may only issue one original of each invoice.
- Any duplicate copies of the invoices issued, under the circumstances provided for, must be marked "duplicate".

6.2 Signing the TBAI invoicing software

This requirement only applies to TBAI invoicing software defined below as desktop applications. It is not a requirement for software defined as distributed architecture applications.

The following definitions will be used for these purposes:

- **Desktop applications:** these are usually distributed on a file over the Internet or in physical form. A physical copy therefore exists on each device using the application.

The software signature will be made on the application installer file (i.e. the file that is distributed), requesting the user's agreement to install the software of the corresponding company. Likewise, the file that launches the application (i.e. the executable file) must also be signed.

- **Applications on distributed architecture:** the most common example of this type of software is probably web applications. In this case, the software is centralised on a server or set of servers. Therefore, no software is distributed and all clients access the same application via a standard light client (for example, a browser).

This type of architecture does not have to comply with the software signature requirements set out in this section.

In the case of desktop applications, the different versions of the invoicing software must be signed. To sign the software, the developer uses a code signing certificate, issued by a recognised certifying authority.

The certificate must display information that will guarantee the identity of the author and the integrity of the software and should be in line with common market practice.

6.3 Application versioning

This requirement only applies to invoicing software defined as desktop applications (see Paragraph 6.2). Applications on distributed architecture (See Paragraph 6.2) will be exempt from the versioning requirements set out in this section, although they must include the Software Version field in the TBAI file.

In desktop applications, the developer must keep a history of all versions and/or updates of the TBAI invoicing software distributed to taxpayers. These versions must be identified by means of a code, which will be indicated in the "Software Version" field in the TBAI file.

These versions of the software must be signed as indicated in Section 6.2 of this document.

The tax authorities may request the version history of the software entered in the TBAI developers register.

6.4 Recommendations on completing the invoice series and number fields

The recommendation for assigning Invoice Series/Numbers is as follows:

Recommended character set: 0123456789ABCDEFGHIJKLMNPQRSTUVWXYZ. To improve readability, the letters I, O and W should be avoided. The use of lower case letters is not recommended.

Blank space: where used, no more than one character should ever be used and the field should never start with a blank space (therefore, text should left-aligned).

Special characters: Hyphen "-", underscore "_", Forward slash "/", dot ".".

6.5 Cancelling an invoice

Should it be necessary to cancel an invoice, signed TBAI files must be generated and sent to the tax authority.

Cancelling an invoice does not require generation of an image.

For cancellation operations, the invoicing software must generate an XML file matching the design and content of the TBAI cancelling format schema. A detailed schema of the TBAI file for cancellation is given in Appendix 2 of this document. In general terms, it must contain the following information:

-
- Subjects: issuer
 - Invoice - Heading: invoice series, invoice number and issue date of the cancelled invoice.
 - TBAI fingerprints:
 - TBAI licence, provided by the TBAI developers register.
 - Developer. Must match the information on the TBAI developers register.
 - Invoicing software: name and version. The software name must match the information in the TBAI developers register.
 - Device. The same considerations apply as for the device field in the TBAI file, as defined in Section 4.1 of this document.

TBAI cancellation files should be signed in the same way as TBAI files, i.e. following the specifications in the “TicketBAI Signature Policy” document.

7 DOCUMENT MANAGEMENT

7.1 Managing specifications and versioning

The provincial governments of Araba/Alava, Bizkaia and Gipuzkoa, and the Basque Government are responsible for maintenance, update, publication and dissemination of this document.

7.2 Signature policy

The specifications for compliance with the requirements of the TicketBAI system are set out in two documents:

- Functional and technical specifications of the TicketBAI system. This document.
- TicketBAI Signature Policy. Describes all specifications relating to the TBAI file signature. This document is available at <https://ticketbai.eus/politicafirma>.

8 APPENDIX 1 – STRUCTURE OF THE TICKETBAI FILE

Legend: * Red = Mandatory fields (In defining mandatory fields, the terms of the SII (<i>Suministro Inmediato de Información</i> or Immediate Information Delivery) have been taken into account) * Black: = Optional Fields * Pink Background = Mutually exclusive fields * Green Background = These fields may be mandatory in some cases, depending on the tax authority in question.					
	Data Block	Data or data group	Format	Possible Values	
TBAI Heading	IDVersionTBAI		Alphanumeric(5)	V 1.0	
Subjects	Issuer	VAT Reg. Number	VatRegNoFormat(9)		
		LAST NAME(S) AND FIRST NAME, COMPANY OR TRADING NAME	Alphanumeric(120)		
	Recipients (1 to 100)	VAT Reg. Number	VatRegNoFormat(9)		
		IDOTRO	COUNTRY CODE	Alphanumeric(2)	
			IDTYPE	Alphanumeric(2)	02: VAT Reg. Number 03: Passport 04: Official ID document issued by country or territory of residence 05: Residence certificate 06: Other identifying document
			ID	Alphanumeric(20)	
		LAST NAME(S) AND FIRST NAME, COMPANY OR TRADING NAME	Alphanumeric(120)		
		POST CODE	Numerical(5)		
		SEVERAL RECIPIENTS	Alphanumeric(1)	S/N. By default, N.	

	Invoice issued by third party or recipient				Alphanumeric(1)	Identifier specifying whether the invoice has been issued by a third party N No. Invoice issued by the issuer. By default, N	
						T Invoice issued by third party.	
						D Invoice issued by the recipient of the transaction.	
Invoice	Invoice Heading	INVOICE SERIES			Alphanumeric(20)		
		INVOICE NUMBER			Alphanumeric(20)		
		INVOICE ISSUE DATE (e.g. 02-12-1992)			DateFormat(10)		
		INVOICE ISSUE TIME (e.g. 21:00:00)			TimeFormat(8)		
		SIMPLIFIED INVOICE			Alphanumeric(1)	S/N. By default, N (Complete invoice).	
		INVOICE ISSUED IN SUBSTITUTION OF SIMPLIFIED INVOICE			Alphanumeric(1)	S/N. By default, N.	
		CORRECTING INVOICE			CODE	Alphanumeric(2)	R1 Correcting Invoice (Error based in law and Art. 80 Paragraphs 1, 2 and 6 of the Provincial VAT Regulation)
							R2 Correcting Invoice (Art. 80.3)
							R3 Correcting Invoice (Art. 80.4)
							R4 Correcting Invoice (Others)
							R5 Correcting Invoice in simplified invoices
					TYPE	Alphanumeric(1)	S For substitution
							I For differences
			CORRECTING REPLACEMENT		CORRECTED BASE	Decimal(12, 2)	

			AMOUNT	CORRECTED TAX AMOUNT	Decimal(12, 2)	
				CORRECTED ADDIT. AMOUNT	Decimal(12, 2)	
		INVOICES CORRECTED OR REPLACED (1 to 100)	INVOICE SERIES		Alphanumerical(20)	
			INVOICE NUMBER		Alphanumerical(20)	
			ISSUE DATE (e.g. 02-12-1992)		DateFormat(10)	
		Invoice data	TRANSACTION DATE (e.g. 02-12-1992)			DateFormat(10)
	INVOICE DESCRIPTION			Alphanumerical(250)		
	INVOICE DATA (1 to 1000)		DETAIL DESCRIPTION		Alphanumerical(250)	
			QUANTITY		Decimal(12.2)	
			UNIT CHARGE		Decimal(12.2)	
			DISCOUNT		Decimal(12.2)	
			TOTAL AMOUNT		Decimal(12.2)	
	TOTAL AMOUNT OF INVOICE			Decimal(12.2)		
	WITHHOLDING			Decimal(12.2)		
	TAX BASE AT COST			Decimal(12.2)		
	CODES OF VAT REGIME CODES AND TAXABLE TRANSACTIONS (1 to 3)			Alphanumerical(2)	01 Transaction in general regime.	
					02 Exports.	
					03 Transactions covered by the special tax regime for used goods, art objects, antiques and collectors' items.	
					04 Special tax regime for investment gold.	

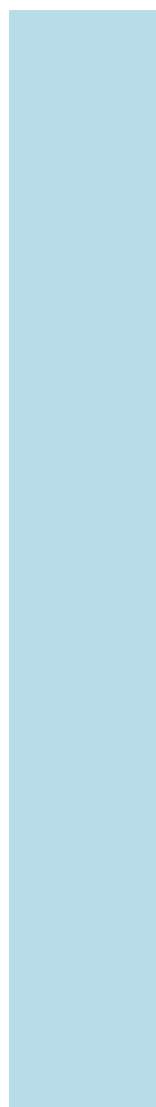
			05 Special regime for travel agencies.
			06 Special regime for consolidated groups in VAT (Advanced Level).
			07 Special cash-basis regime.
			08 Transactions subject to Tax on Production, Services and Imports / General Indirect Tax of the Canary Islands).
			09 Sales from provision of services by travel agencies acting as mediators on behalf of third parties. (Additional Provision 3 of the Invoicing Regulations)
			10 Amounts collected on behalf of third parties of professional fees or rights derived from industrial property, copyright or others on behalf of members or associates made by societies, associations, professional institutes or other entities performing these payment collection functions.

						11 Transactions involving rental of business premises subject to withholding.
						12 Transactions involving rental of business premises not subject to withholding.
						13 Transactions involving rental of business premises subject or not to withholding.
						14 Invoice with VAT pending accrual in site certifications whose recipient is a public administration (government authority).
						15 Invoice with VAT pending accrual in chain of title transactions.
						51 Transactions in special additional VAT regime.
						52 Transactions in simplified regime.
	Itemisation at invoice level* (Invoice itemisation: When the other party is “national” / itemisation of type of transaction: Delivery of goods / Provision of service: When the other party is non-national)					
	Itemisation at invoice level* (Invoice itemisation: When the other party is “national” or there is no other party)	SUBJECT	EXEMPT ExemptDetail (1 to 7, a grouping of data by grounds for exemption)	CAUSE	Alphanumerical(2)	E1 EXEMPT Art. 20
						E2 EXEMPT Art. 21
E3 EXEMPT Art. 22						
E4 EXEMPT Art. 23 and 24						
E5 EXEMPT Art. 25						

								E6 EXEMPT Others
								EXEMPT TAX BASE
								Decimal(12.2)
								NON EXEMPT RATE
								Alphanumeric(2)
								S1 without reverse charge
								S2 with reverse charge
								TAX BASE
								Decimal(12.2)
								TAX RATE
								Decimal(3.2)
								TAX AMOUNT
								Decimal(12.2)
								ADDITIONAL VAT RATE
								Decimal(3.2)
								ADDITIONAL VAT AMOUNT
								Decimal(12.2)
								AMOUNTS VATDetail (1 to 6, a grouping of data by type)
								TRANSACTION IN ADDITIONAL VAT OR SIMPLIFIED REGIME
								Alphanumeric(1)
								Transaction in Additional VAT or simplified regime The default value is "N". If marked "S", the invoice is issued by a taxpayer in simplified or additional VAT regime.

			NON-TAXABLE	GROUNDS FOR NON-TAXABILITY		Alphanumeric(2)	OT Non-taxable under Art. 7, 14, Others
			Non-TaxableDetail (1 to 2)				RL Non-taxable under location rules.
				AMOUNT	Decimal(12.2)		
	Itemisation at transaction level (When the other party is non-national)	Itemisation at level of service provision	SUBJECT	EXEMPT ExemptDetail (1 to 7, a grouping of data by grounds for exemption)	CAUSE	Alphanumeric(2)	E1 EXEMPT Art. 20 E2 EXEMPT Art. 21 E3 EXEMPT Art. 22 E4 EXEMPT Art. 23 and 24 E5 EXEMPT Art. 25 E6 EXEMPT Others
					EXEMPT TAX BASE	Decimal(12.2)	
				NOT EXEMPT (1 to 2)	NON EXEMPT RATE	Alphanumeric(2)	S1 without reverse charge S2 with reverse charge
					AMOUNTS VATDetail (1 to 6, a grouping of data by type)	TAX BASE	Decimal(12.2)
						TAX RATE	Decimal(3.2)
						TAX AMOUNT	Decimal(12.2)
						ADDITIONAL VAT RATE	Decimal(3.2)
						ADDITIONAL VAT AMOUNT	Decimal(12.2)

				TRANSACTION IN ADDITIONAL VAT OR SIMPLIFIED REGIME	Alphanumeric(1)	Transaction in Additional VAT or Simplified regime The default value is "N". If marked "S", the invoice is issued by a taxpayer in simplified regime or additional VAT regime.
				NON-TAXABLE Non-TaxableDetail (1 to 2)	GROUND FOR NON-TAXABILITY	Alphanumeric(2) OT Non-taxable under Art. 7, 14, Others RL Non-taxable under location rules.
Itemisation at level of delivery of goods	SUBJECT	EXEMPT ExemptDetail (1 to 7, a grouping of data by grounds for exemption)		AMOUNT	Decimal(12.2)	
				CAUSE	Alphanumeric(2)	E1 EXEMPT Art. 20 E2 EXEMPT Art. 21 E3 EXEMPT Art. 22 E4 EXEMPT Art. 23 and 24 E5 EXEMPT Art. 25 E6 EXEMPT Others
				EXEMPT TAX BASE	Decimal(12.2)	
				NOT EXEMPT (1 to 2)	NON EXEMPT RATE	Alphanumeric(2)



AMOUNTS VATDetail (1 to 6, a grouping of data by type)	TAX BASE	Decimal(12.2)	
		TAX RATE	Decimal(3.2)
		TAX AMOUNT	Decimal(12.2)
	ADDITIONAL VAT RATE	Decimal(3.2)	
		ADDITIONAL VAT AMOUNT	Decimal(12.2)
	TRANSACTION IN ADDITIONAL VAT OR SIMPLIFIED REGIME	Alphanumeric(1)	Transaction in Additional VAT or Simplified regime The default value is "N". If marked "S", the invoice is issued by a taxpayer in simplified regime or additional VAT regime.
NON-TAXABLE Non-TaxableDetail (1 to 2)	GROUNDS FOR NON-TAXABILITY	Alphanumeric(2)	OT Non-taxable under Art. 7, 14, Others RL Non-taxable under location rules
		AMOUNT	Decimal(12.2)

Fingerprints	TicketBAI fingerprints		Chaining, previous invoice	Series, previous invoice		Alphanumeric(20)	
				Invoice Number, previous invoice		Alphanumeric(20)	
				Issue Date, previous invoice		DateFormat(10)	
				Signature, previous invoice		Alphanumeric(100)	First hundred characters of the SignatureValue field of the TBAI file of the previous invoice
			Invoicing software	TBAI Licence		Alphanumeric(20)	
				VAT Reg. Number of developer		VatRegNoFormat(9)	Datum associated with TBAI Licence.
				Developer ID, other	COUNTRY CODE	Alphanumeric(2)	Datum associated with TBAI Licence.
					IDTYPE	Alphanumeric(2)	Datum associated with TBAI Licence.
					ID	Alphanumeric(20)	Datum associated with TBAI Licence.
				Name of Software		Alphanumeric(120)	Datum associated with TBAI Licence.
				Software Version		Alphanumeric(20)	
			Device	Serial No. of invoicing device		Alphanumeric(30)	
Invoice Signature	Electronic Signature		Signature Data	TBAI signature		See “TicketBAI Signature Policy”	

9 APPENDIX 2 – STRUCTURE OF THE TICKETBAI CANCELLATION FILE

Legend: * Red = Mandatory fields (In defining mandatory fields, the terms of the SII (Suministro Inmediato de Información or Immediate Information Delivery) have been taken into account) * Black: = Optional Fields * Pink Background = Mutually exclusive fields						
	Data Block	Data or data group		Format	Possible Values	
TBAI Heading	IDVersionTBAI			Alphanumeric(5)	V 1.0	
Invoice ID	Issuer	VAT Reg. Number		VatRegNoFormat(9)		
		LAST NAME(S) AND FIRST NAME, COMPANY OR TRADING NAME		Alphanumeric(120)		
	Invoice Heading	CANCELLED INVOICE SERIES		Alphanumeric(20)		
		CANCELLED INVOICE NUMBER		Alphanumeric(20)		
		CANCELLED INVOICE ISSUE DATE (e.g. 02-12-1992)		DateFormat(10)		
Fingerprints	TicketBAI fingerprints	Invoicing software	TBAI Licence	Alphanumeric(20)		
			VAT Reg. Number of developer		VatRegNoFormat(9)	Datum associated with TBAI Licence.
			Developer ID, other	COUNTRY CODE	Alphanumeric(2)	Datum associated with TBAI Licence.
				IDTYPE	Alphanumeric(2)	Datum associated with TBAI Licence.
				ID	Alphanumeric(20)	Datum associated with TBAI Licence.
			Name of Software		Alphanumeric(120)	Datum associated with TBAI Licence.
			Software version		Alphanumeric(20)	
		Device	Serial No. of invoicing device	Alphanumeric(30)		

Invoice Signature	Electronic Signature	Signature Data	TBAI signature	See "TicketBAI Signature Policy"
-------------------	----------------------	----------------	----------------	----------------------------------

10 APPENDIX 3 – ERROR-CHECKING CRC ALGORITHM

```
package net.ticketbai;

import java.io.UnsupportedEncodingException;

public class CRC8{

    private static final String UTF_8 = "UTF-8";

    // CRC-8,
    // Poly = x^8 + x^2 + x^1 + x^0
    // Init = 0
    // RefIn = false
    // RefOut = false
    // XorOut = 0
    private static final byte[] crc8_table =
        new byte[] {
            (byte) 0x00, (byte) 0x07, (byte) 0x0E, (byte) 0x09, (byte) 0x1C, (byte) 0x1B, (byte) 0x12, (byte) 0x15,
            (byte) 0x38, (byte) 0x3F, (byte) 0x36, (byte) 0x31, (byte) 0x24, (byte) 0x23, (byte) 0x2A, (byte) 0x2D,
            (byte) 0x70, (byte) 0x77, (byte) 0x7E, (byte) 0x79, (byte) 0x6C, (byte) 0x6B, (byte) 0x62, (byte) 0x65,
            (byte) 0x48, (byte) 0x4F, (byte) 0x46, (byte) 0x41, (byte) 0x54, (byte) 0x53, (byte) 0x5A, (byte) 0x5D,
            (byte) 0xE0, (byte) 0xE7, (byte) 0xEE, (byte) 0xE9, (byte) 0xFC, (byte) 0xFB, (byte) 0xF2, (byte) 0xF5,
            (byte) 0xD8, (byte) 0xDF, (byte) 0xD6, (byte) 0xD1, (byte) 0xC4, (byte) 0xC3, (byte) 0xCA, (byte) 0xCD,
            (byte) 0x90, (byte) 0x97, (byte) 0x9E, (byte) 0x99, (byte) 0x8C, (byte) 0x8B, (byte) 0x82, (byte) 0x85,
            (byte) 0xA8, (byte) 0xAF, (byte) 0xA6, (byte) 0xA1, (byte) 0xB4, (byte) 0xB3, (byte) 0xBA, (byte) 0xBD,
            (byte) 0xC7, (byte) 0xC0, (byte) 0xC9, (byte) 0xCE, (byte) 0xDB, (byte) 0xDC, (byte) 0xD5, (byte) 0xD2,
            (byte) 0xFF, (byte) 0xF8, (byte) 0xF1, (byte) 0xF6, (byte) 0xE3, (byte) 0xE4, (byte) 0xED, (byte) 0xEA,
            (byte) 0xB7, (byte) 0xB0, (byte) 0xB9, (byte) 0xBE, (byte) 0xAB, (byte) 0xAC, (byte) 0xA5, (byte) 0xA2,
            (byte) 0x8F, (byte) 0x88, (byte) 0x81, (byte) 0x86, (byte) 0x93, (byte) 0x94, (byte) 0x9D, (byte) 0x9A,
            (byte) 0x27, (byte) 0x20, (byte) 0x29, (byte) 0x2E, (byte) 0x3B, (byte) 0x3C, (byte) 0x35, (byte) 0x32,
            (byte) 0x1F, (byte) 0x18, (byte) 0x11, (byte) 0x16, (byte) 0x03, (byte) 0x04, (byte) 0x0D, (byte) 0x0A,
            (byte) 0x57, (byte) 0x50, (byte) 0x59, (byte) 0x5E, (byte) 0x4B, (byte) 0x4C, (byte) 0x45, (byte) 0x42,
            (byte) 0x6F, (byte) 0x68, (byte) 0x61, (byte) 0x66, (byte) 0x73, (byte) 0x74, (byte) 0x7D, (byte) 0x7A,
            (byte) 0x89, (byte) 0x8E, (byte) 0x87, (byte) 0x80, (byte) 0x95, (byte) 0x92, (byte) 0x9B, (byte) 0x9C,
            (byte) 0xB1, (byte) 0xB6, (byte) 0xBF, (byte) 0xB8, (byte) 0xAD, (byte) 0xAA, (byte) 0xA3, (byte) 0xA4,
            (byte) 0xF9, (byte) 0xFE, (byte) 0xF7, (byte) 0xF0, (byte) 0xE5, (byte) 0xE2, (byte) 0xEB, (byte) 0xEC,
            (byte) 0xC1, (byte) 0xC6, (byte) 0xCF, (byte) 0xC8, (byte) 0xDD, (byte) 0xDA, (byte) 0xD3, (byte) 0xD4,
            (byte) 0x69, (byte) 0x6E, (byte) 0x67, (byte) 0x60, (byte) 0x75, (byte) 0x72, (byte) 0x7B, (byte) 0x7C,
            (byte) 0x51, (byte) 0x56, (byte) 0x5F, (byte) 0x58, (byte) 0x4D, (byte) 0x4A, (byte) 0x43, (byte) 0x44,
            (byte) 0x19, (byte) 0x1E, (byte) 0x17, (byte) 0x10, (byte) 0x05, (byte) 0x02, (byte) 0x0B, (byte) 0x0C,
            (byte) 0x21, (byte) 0x26, (byte) 0x2F, (byte) 0x28, (byte) 0x3D, (byte) 0x3A, (byte) 0x33, (byte) 0x34,
            (byte) 0x4E, (byte) 0x49, (byte) 0x40, (byte) 0x47, (byte) 0x52, (byte) 0x55, (byte) 0x5C, (byte) 0x5B,
            (byte) 0x76, (byte) 0x71, (byte) 0x78, (byte) 0x7F, (byte) 0x6A, (byte) 0x6D, (byte) 0x64, (byte) 0x63,
            (byte) 0x3E, (byte) 0x39, (byte) 0x30, (byte) 0x37, (byte) 0x22, (byte) 0x25, (byte) 0x2C, (byte) 0x2B,
            (byte) 0x06, (byte) 0x01, (byte) 0x08, (byte) 0x0F, (byte) 0x1A, (byte) 0x1D, (byte) 0x14, (byte) 0x13,
            (byte) 0xAE, (byte) 0xA9, (byte) 0xA0, (byte) 0xA7, (byte) 0xB2, (byte) 0xB5, (byte) 0xBC, (byte) 0xBB,
            (byte) 0x96, (byte) 0x91, (byte) 0x98, (byte) 0x9F, (byte) 0x8A, (byte) 0x8D, (byte) 0x84, (byte) 0x83,
            (byte) 0xDE, (byte) 0xD9, (byte) 0xD0, (byte) 0xD7, (byte) 0xC2, (byte) 0xC5, (byte) 0xCC, (byte) 0xCB,
            (byte) 0xE6, (byte) 0xE1, (byte) 0xE8, (byte) 0xEF, (byte) 0xFA, (byte) 0xFD, (byte) 0xF4, (byte) 0xF3 };

    /**
     * Calculate the CRC value with data from input string.
     * @param input input string
     * @return The calculated CRC value. Left padding with zeros.
     * @throws UnsupportedOperationException
     */
    public static String calculate(String input) throws UnsupportedOperationException {
        byte[] data = input.getBytes(UTF_8);
        int len = data.length;
        byte crc = 0;

        for (int i = 0; i < len; i++)
            crc = crc8_table[(crc ^ data[i]) & 0xff];

        return String.format("%03d", crc & 0xFF);
    }
}
```