Guida alla Configurazione

Model

In the process of configuring a robust and versatile system for managing clinical visits, it is crucial to understand the architecture of the involved classes. This model is based on a clear and interconnected organization of four fundamental elements: **Struttura Visita**, **Agenda**, **Tipi**, and **Viste**.

Struttura Visita is the core of the System, linking and coordinating the other components. A Struttura Visita is associated with a specific *Tipo* (Type) that precisely defines the data structure used to represent *Clinical Visits*.

Agenda defines the clinical studies associated with each Struttura Visita. This part of the system allows for the organization of visits based on specific clinical or research contexts, offering a structured and orderly view of the scheduled activities.

Tipo defines the data structure in order to accurately represent information related to Clinical Visits. Each *Struttura Visita* is associated with a specific *Tipo* (Type) that determines the way of managing and storing such structured data, allowing a detailed and precise representation of the information collected during the clinical visit.

Lastly, **Viste** *(Viewers)* are the tools used to view and interact with the data defined by the *Tipi* (Types). These components provide an intuitive and customizable user interface to access and interpret information related to Clinical Visits, enabling users to efficiently and accurately view and analyze the stored data.

Tipi e **Viste** are defined using a domain-specific language (DSL), a programming language that mimics the terms, idioms, and expressions used by experts in a particular domain.

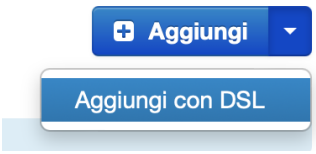
These data structures can be defined through an interface that allows the insertion, modification, and deletion of *Types* and *Viewers*. When a *Type* or a *Viewer* is defined, it can be validated or previewed before saving, to ensure it is correct.

Viewers' templates consist of both DSL and CSS, to customize the display of data contained in the structures defined by *Types*. These templates can be configured in a manner similar to that described for the *Types*, but there are also scripts available for their automatic generation based on the *Types*.

The following sections illustrate how to insert each of these elements, starting from *Types* up to the *Struttura Visita*, which will gather all the previously defined information, linking them together and defining the structure of the *Clinical Visit*.

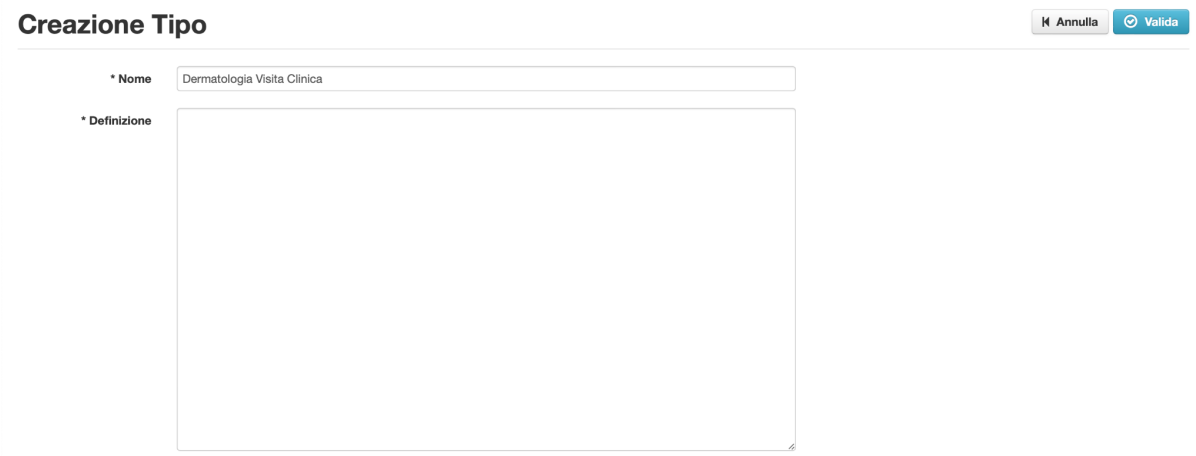
Type Definition

The Type can be added by navigating to the "Configurazione" tab, then "Tipi".

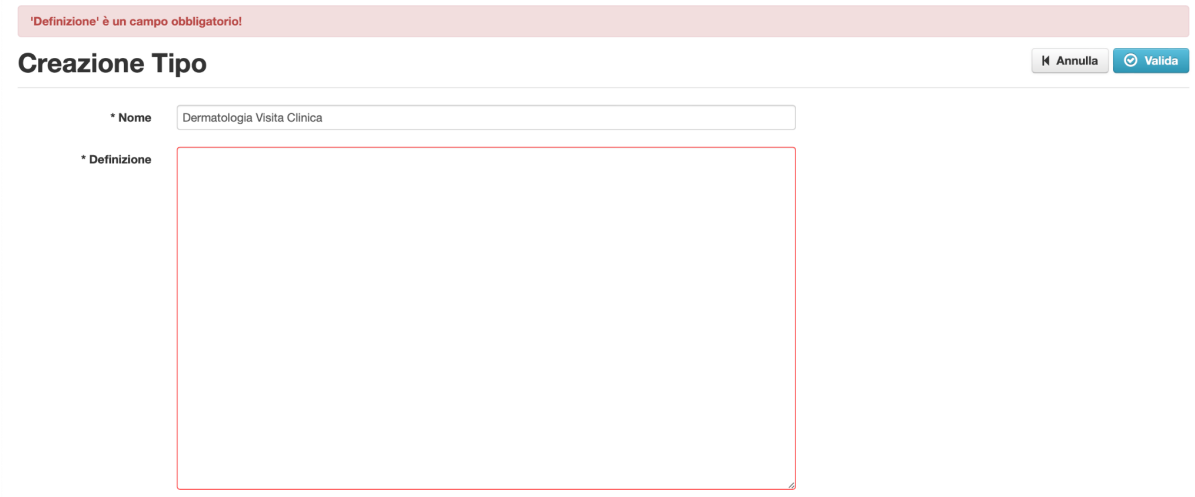


This will open a list of all the Types already present. A new Type can be added by going to "Aggiungi" (Add), then "Aggiungi con DSL" (Add with DSL).

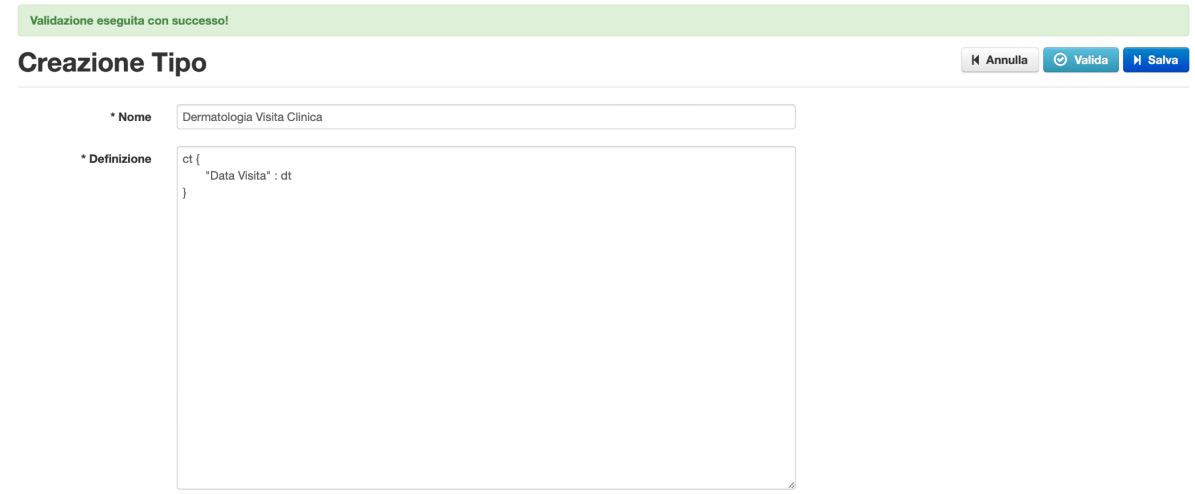
At this point, the "Create Type" screen will open with two main fields: "Nome" (Name) and "Definizione" (Definition).



Both fields are necessary for the definition of the Type, and the Name must be unique. To save a Type, it must first be validated by clicking on "Valida" (Validate). If you try to validate a Type without specifying its definition, an error message will be displayed.



By adding a simple definition (more detailed explanations will be provided below), you can see how the validation is successfully carried out, allowing the Type to be saved.



Within the "Definition" field, the structure of a type can be defined as if it were a tree. Below are the elements that can be defined:

- **ct** *composite type*, which can contain other elements inside. They can be nested and group other elements within them. The syntax with which they are defined is as follows: (“Nome_Composite” can also be omitted)

```
["Nome_Composite"]: ct {...}
```

- **qt** *quantitative type,* is defined by specifying two values: unit of measurement and a pair of values (integer, decimal). For example:

```
"Quante sigarette al giorno" : qt {"(Indefinito)"(2,0)},
"Da quanto ha smesso di fumare" : qt{"mese"(2,0)}
```

- **ql** *qualitative type*, is defined by specifying a list of value strings and -optionally- "ordered". For example:

```
"Diagnosi" : ql ordered { "Nevo Comune",
                          "Nevo Atipico",
```

```
                                        "Melanoma",
                                        "Epitelioma"},
            "Fumo": ql ordered {"Sì", "No", "Ex fumatore"},
```

- **dt** *temporal type*, used to define dates. For example:

```
           "Data Visita" : dt
```

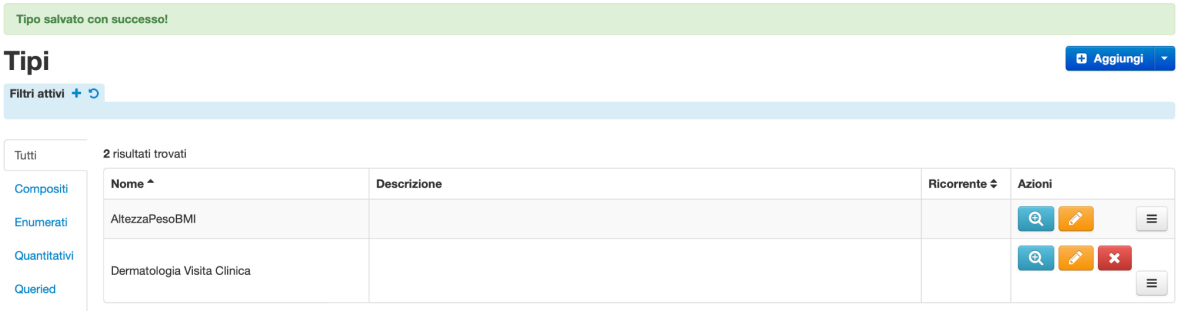
- **tx** *textual type,* used to specify textual fields. For example:

```
           "Conclusioni" : tx
```

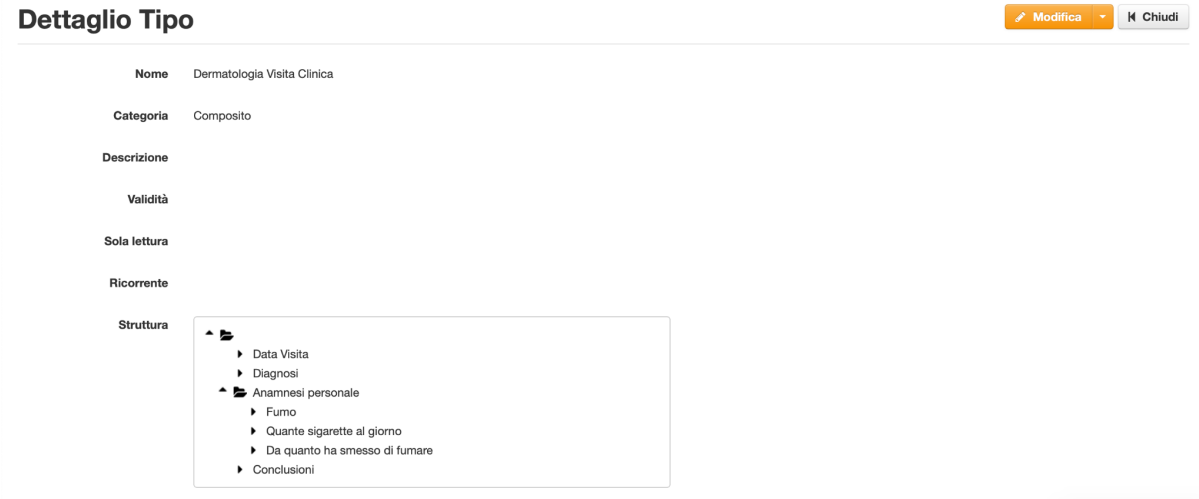
Summarizing all the various fields and collecting them into a single composite type, a tree structure can then be obtained, as in the example Type below.

```
ct {

       "Data Visita" : dt,
 "Diagnosi" : ql ordered { "Nevo Comune",

              "Nevo Atipico",
               "Melanoma",
               "Epitelioma"},

"Anamnesi personale": ct {
            "Fumo": ql ordered {"Sì", "No", "Ex fumatore"},
            "Quante sigarette al giorno" : qt
{"(Indefinito)"(2,0)},
            "Da quanto ha smesso di fumare" : qt{"mese"(2,0)}
},
"Conclusioni" : tx
}
```

Once saved, the "Dermatologia Visita Clinica" Type will be viewable in the list of Types under Configurazione->Tipi (Configuration->Types).



By clicking on the magnifying glass, you can see the details of the Type, its structure, and the fields it contains.

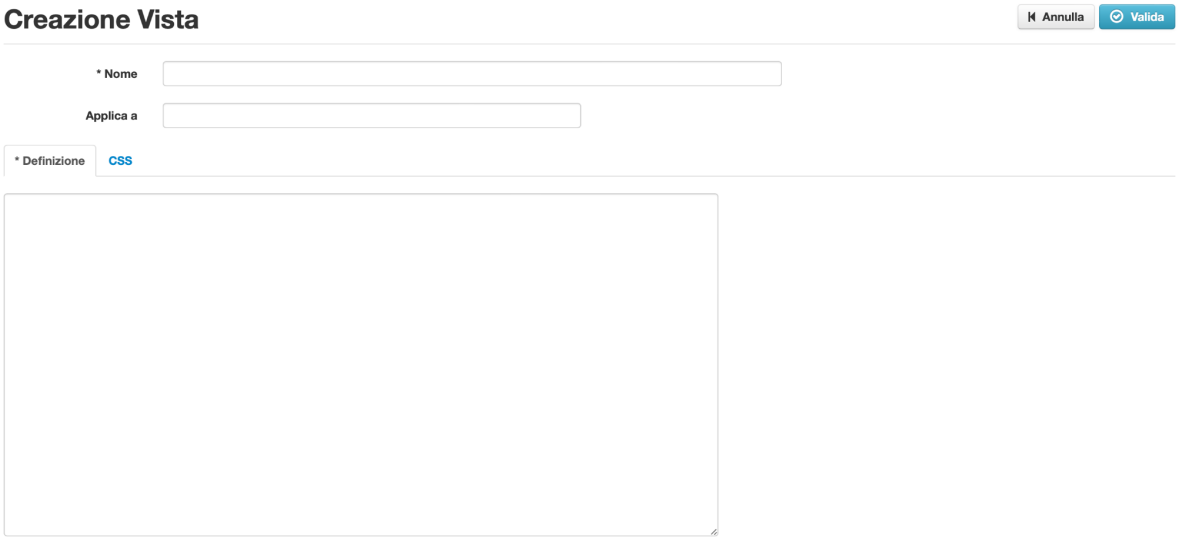


Viewer Definition

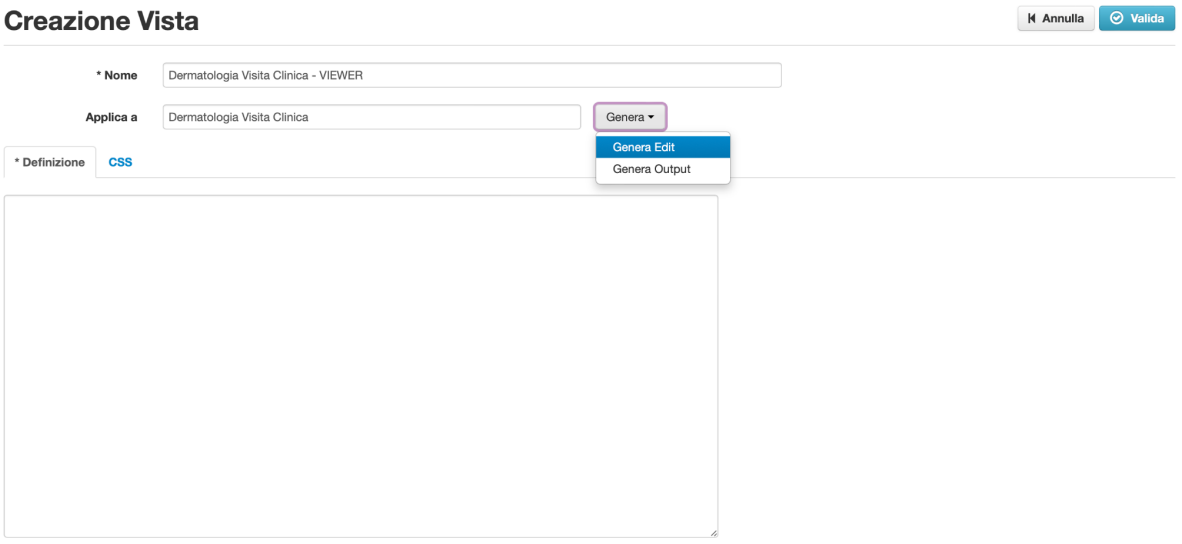
Once the Type is defined, it is possible to generate a basic structure of viewers, according to needs. Viewers are distinguished into "Viewer Output" or "Viewer Edit". “Viewer Output” does not allow for the specification of fields, they display what has already been written within the structure of the visit. On the other hand, “Viewer Edit” are used when the structure of the visit is being compiled, and each field will have its own appropriately associated element.

The Viewer can be added by navigating to the “Configurazione” (“Configuration") tab, then “Viste”("Viewers").

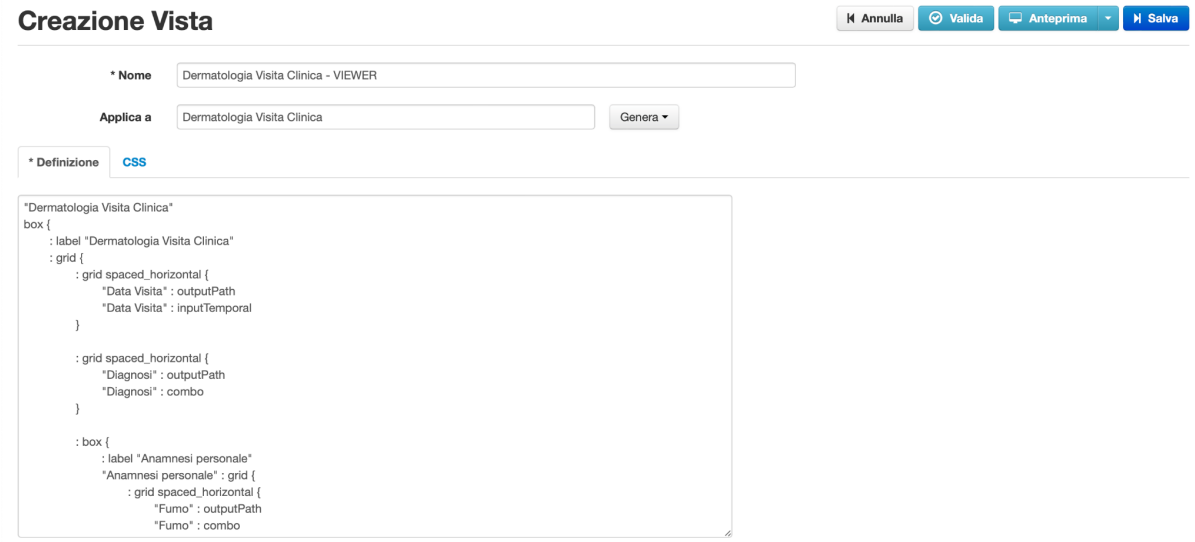
After that, a list of all the existing *Viewers* will open. A new Viewer can be added by clicking on “Aggiungi” ("Add").



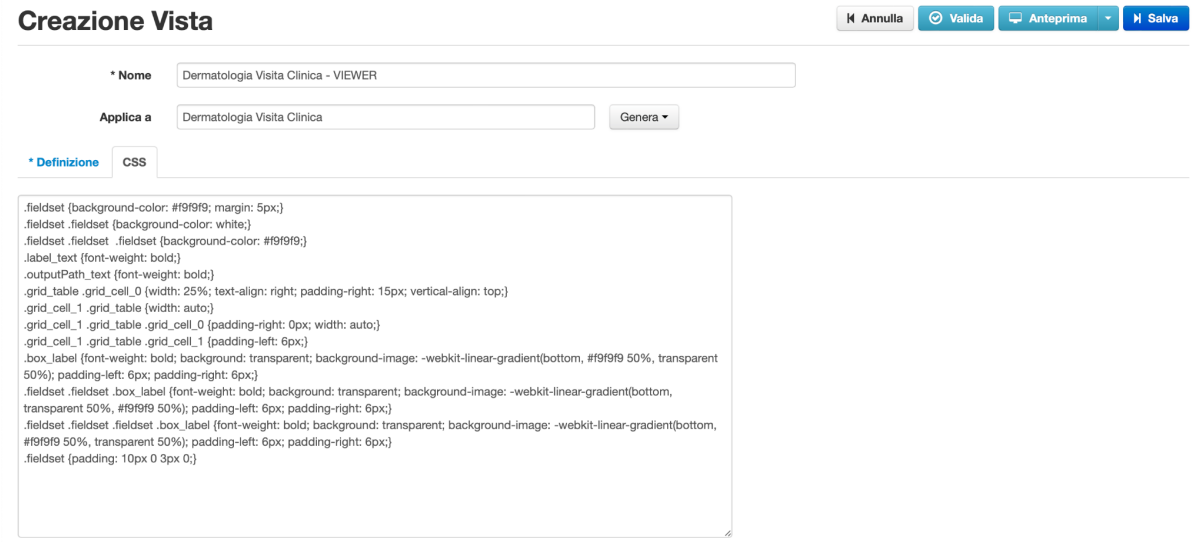
Creating a Viewer involves specifying a Name, a Definition, and a Type to apply it to. Once the Type of the Viewer is specified, the interface allows for the automatic generation of Edit or Output Viewers, and related CSS, via the “Genera” ("Generate") button.



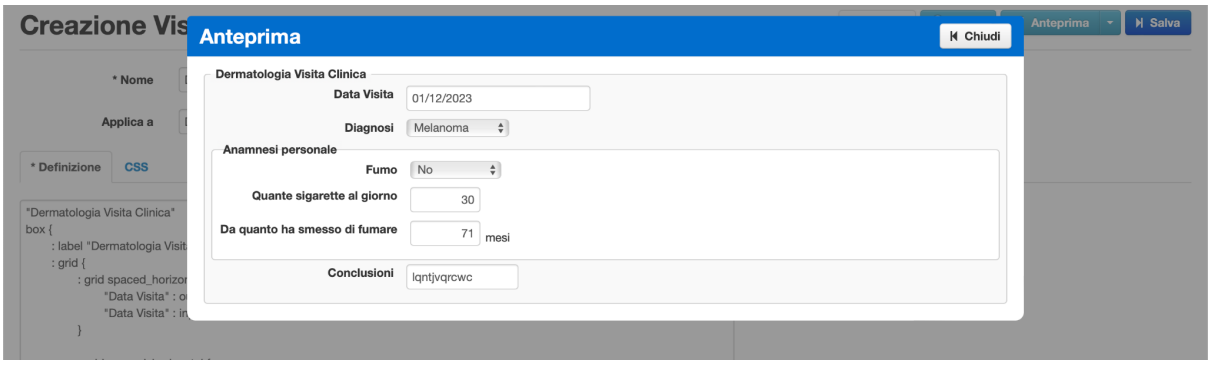
Below, we can see the result applied to the *Viewer Edit* of the previously defined Type.



A default definition of the Viewer's CSS is also provided, which can be further customized if necessary.



Using the “Anteprima” ("Preview") button at the top, it is also possible to see how the fields will be displayed, filled with random data.



The structure of a Viewer is composed of *grid* and *box* elements nested within each other. In case of more extensive Clinical Folders, It is also possible to have a *tabbedPanel*, the outermost element, to specify a tab structure.

In the following example It is possible to see the **Viewer Edit** of the previously defined Type “Dermatologia Clinica”.

```
"Dermatologia Visita Clinica"
box {
	: label "Dermatologia Visita Clinica"
	: grid {
		: grid spaced_horizontal {
			"Data Visita" : outputPath
			"Data Visita" : inputTemporal
		}
		: grid spaced_horizontal {
			"Diagnosi" : outputPath
			"Diagnosi" : combo
		}
		: box {
			: label "Anamnesi personale"
			"Anamnesi personale" : grid {
				: grid spaced_horizontal {
					"Fumo" : outputPath
					"Fumo" : combo
				}
				: conditionalPanel {
				"Fumo" : "Sì"
				 clear "Quante sigarette al giorno"
				: grid spaced_horizontal {
					"Quante sigarette al giorno" : outputPath
					"Quante sigarette al giorno" : inputText
					}
				}
				: conditionalPanel {
				"Fumo" : "Ex fumatore"
				clear "Da quanto ha smesso di fumare"
				: grid spaced_horizontal {
					"Da quanto ha smesso di fumare" : outputPath
					"Da quanto ha smesso di fumare"
					: grid spaced_horizontal {
								: inputText
								: combo
								}
					}
				}
			}
		}
		: grid spaced_horizontal {
			"Conclusioni" : outputPath
			"Conclusioni" : textArea
		}
	}
}
```

Below, the Viewer Output

```
"Dermatologia Visita Clinica"
box {
	: label "Dermatologia Visita Clinica"
	: grid {
		: grid spaced_horizontal {
```

```
                        "Data Visita" : outputPath
                        "Data Visita" : outputValue
                }
                : grid spaced_horizontal {
                        "Diagnosi" : outputPath
                        "Diagnosi" : outputValue
                }
                : box {
                        : label "Anamnesi personale"
                        "Anamnesi personale" : grid {
                                : grid spaced_horizontal {
                                        "Fumo" : outputPath
                                        "Fumo" : outputValue
                                }
                                        : conditionalPanel {
                                "Fumo" : "Sì"
                                 clear "Quante sigarette al giorno"
                                : grid spaced_horizontal {
                                                "Quante sigarette al giorno" : outputPath
                                                "Quante sigarette al giorno" : outputValue
                                        }
                                        }
                                        : conditionalPanel {
                                        "Fumo" : "Ex fumatore"
                                        clear "Da quanto ha smesso di fumare"
                                        : grid spaced_horizontal {
                                                        "Da quanto ha smesso di fumare" :
outputPath
                                                        "Da quanto ha smesso di fumare" : grid
spaced_horizontal {
                                                        : outputValue
                                                        : outputValue
                                                        }
                                        }
                                        }
                        }
                }
                : grid spaced_horizontal {
                        "Conclusioni" : outputPath
                        "Conclusioni" : outputValue
                }
        }
}
```

The 'final' elements change from edit to output. If the viewer being constructed is an edit viewer, a corresponding element must be placed for the field that is to be filled out. Therefore, we will have:

- *textArea / inputText* if the specified field is of type :tx. TextArea is used to handle larger textual fields, such as the conclusions of a Clinical Visit.
- *combo* if it is of type :ql or :qt
- *inputTemporal* if it is of type :dt.

In the case of a Viewer Output, however, we will always have the following structure:

```
"Label": outputPath
"Value": outputValue
```

Unlike the auto-generated Viewers, the ones described previously specify **conditionalPanels**. *ConditionalPanels* are elements, used in combination with **'ql qualitative types'**, that allow for the management of the appearance of specific forms when a particular value is entered in the qualitative type. Specifically, we can see in the provided example, that depending on the possible responses to the field “Fumo?” (Sì/Ex Fumatore/No), different fields appear:

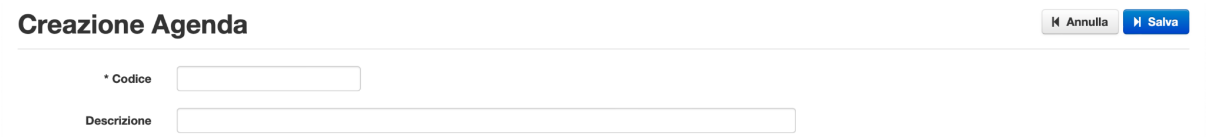
- If the answer is Sì, it is required to specify the number of cigarettes per day, and thus the appropriate form appears: “Quante sigarette al giorno” ('How many cigarettes per day?'), while “Da quanto ha smesso di fumare” ('How long since you quit smoking') remains hidden and unfillable.
- If the answer is “Ex Fumatore” (Ex Smoker), the opposite effect occurs: the form related to “Quante sigarette al giorno” ('How many cigarettes per day?') is hidden, and “Da quanto ha smesso di fumare” ('How long since you quit smoking') appears instead.
- If the answer is 'No', there is no need to specify anything else, and both fields remain hidden.

Agenda

Agenda can be added by navigating to the “Configurazione” tab (Configuration), then “Agende” (Agendas).

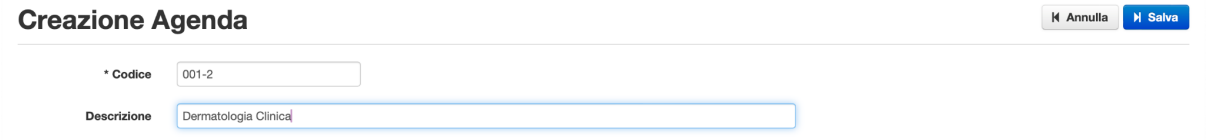
After that, a list of all the existing Agendas will open. A new Agenda can be added by clicking on “Aggiungi” ('Add').

This will open a “Creazione Agenda” ('Create Agenda') window that will allow for the definition of an Agenda.



To define an Agenda, it is necessary to specify a code (unique) and a description.

We can fill in the fields as follows:



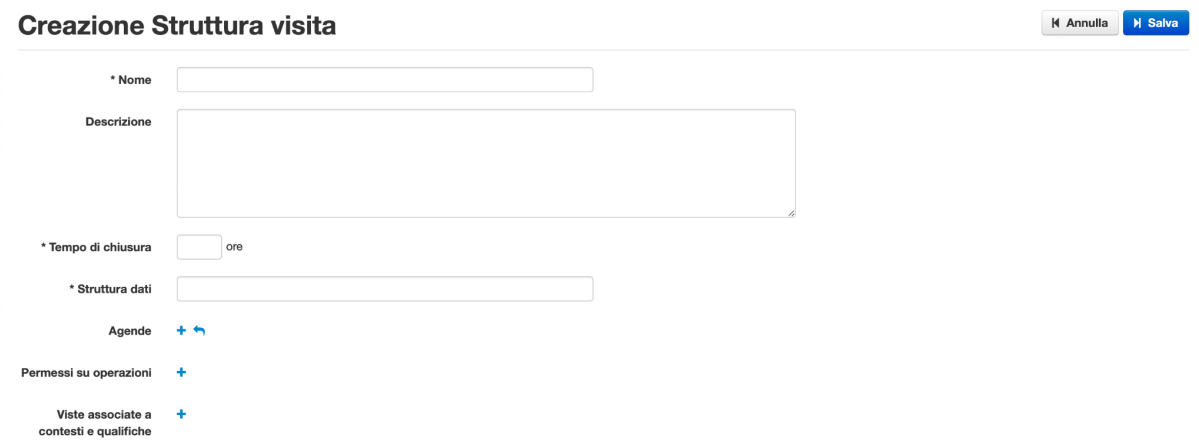
Struttura Visita

Once the Types, Viewers, and Agenda have been added, it is possible to gather all this information together by specifying a *Struttura Visita*

A Visit Structure can be added by navigating to the "Configurazione" tab (‘Configuration’), then "Strutture visita" (‘Visit Structures’).

After that, a list of all the existing *Strutture Visita* will open. A new *Struttura Visita* can be added by clicking on “Aggiungi” (‘Add’).

This will open the usual creation screen that we have already seen for the previous elements.



To define a Struttura Visita, it is necessary to specify:

- *Nome*, unique, of the Struttura Visita.
- *Descrizione* (optional) explanatory.
- *Tempo di chiusura automatico delle visite* (Automatic closure time of visits): if a Visit is not explicitly concluded by the doctor, this field specifies after how much time It will be automatically closed and considered no longer valid.
- *Struttura Dati*: the Type defined previously.
- *Agende*: the agenda(s) defined previously.
- *Permessi su operazioni* (Permissions on operations): user authorizations regarding certain operations.
- *Viste associate a contesti e qualifiche* (Viewers associated with contexts and qualifications): here will be listed the Viewers previously created, they will also be associated with users authorized to compile, manage, and view them. It will be specified in what circumstances one or the other viewer will be used.

Based on the data specified previously in this guide, we can compile the *Struttura Visita* in the following way: