**Estructura del Código fuente del Aplicativo**

El marco de trabajo de desarrollo organiza la aplicación en directorios de configuración y ubicación del código fuente bajo la siguiente estructura de árbol:

**/valueme (aplicación web)**

├── gradle

│ └── wrapper

├── **grails-app**

│ ├── **assets**

│ │ ├── images

│ │ ├── javascripts

│ │ └── stylesheets

│ │ └── themes

│ │ └── default

│ │ └── assets

│ │ ├── fonts

│ │ └── images

│ ├── **conf**

│ │ └── spring

│ ├── **controllers**

│ │ └── valueme

│ ├── **domain**

│ │ └── valueme

│ ├── **i18n**

│ ├── **init**

│ │ └── valueme

│ ├── **services**

│ │ └── valueme

│ ├── **taglib**

│ │ └── valueme

│ └── **views**

│ ├── appInfo

│ ├── assessment

│ ├── category

│ ├── categoryType

│ ├── customField

│ ├── emails

│ ├── layouts

│ ├── login

│ ├── meci

│ ├── param

│ ├── permission

│ ├── progress

│ ├── question

│ ├── role

│ ├── schedule

│ ├── score

│ ├── survey

│ └── userAccount

**Directorios Principales:**

**grails-app:** Contiene el directorio principal del código fuente de desarrollo de propósito específico.

**assets:** En este directorio se almacena el código fuente que gestionan los estilos visuales CSS y el código en Javascript requerido en la aplicación:

* Estilos CSS
* Librería Semantic UI
* Librería Google Charts
* Librerías de AngularJS, Jquery.

**├── images**

**│ ├── favicon.ico**

**│ ├── fondo\_idipron.png**

**│ ├── meci.png**

**│ ├── paragraph.png**

**│ └── processmap.png**

**├── javascripts**

**│ ├── analyticstracking.js**

**│ ├── angular-app.js**

**│ ├── angular.min.js**

**│ ├── angular-resource.min.js**

**│ ├── angular-route.min.js**

**│ ├── application.js**

**│ ├── calendar.js**

**│ ├── charts.js**

**│ ├── jquery-2.2.0.min.js**

**│ ├── jquery.gfeed.js**

**│ ├── jquery.maphilight.js**

**│ ├── semantic.min.js**

**│ └── tbl-cstom-field.js**

**└── stylesheets**

**├── application.css**

**├── calendar.css**

**├── errors.css**

**├── semantic.min.css**

**conf:** En este directorio se configuran los parámetros de la aplicación mediante archivos de configuración:

* Spring Security.
* Servicio de correo electrónico.
* Configuración de la base de datos.
* Configuración de ambientes de producción y pruebas.
* Parámetros de sesión.

**├── application.groovy**

**├── application.yml**

**├── logback.groovy**

**└── spring**

**└── resources.groovy**

**controllers:** En este directorio se gestionan todos los controladores de la arquitectura MVC, cada controlador proporciona acceso a los servicios y objetos de dominio, los controladores se encargan de procesar las peticiones del navegador web.

* Controladores:

**├── AppInfoController.groovy**

**├── AssessmentController.groovy**

**├── CategoryController.groovy**

**├── CategoryTypeController.groovy**

**├── CustomFieldController.groovy**

**├── MeciController.groovy**

**├── ParamController.groovy**

**├── PermissionController.groovy**

**├── ProgressController.groovy**

**├── QuestionController.groovy**

**├── RoleController.groovy**

**├── ScheduleController.groovy**

**├── ScoreController.groovy**

**├── SurveyController.groovy**

**├── UrlMappings.groovy**

**└── UserAccountController.groovy**

**domain:** En este directorio contiene todas las clases de dominio de la aplicación definidas en el modelo de dominio.

* Cada clase de dominio puede tener relaciones de asociación con otras.
* Las clases de dominio contienen configuraciones de validaciones de datos del lado del servidor.
* Cada clase de dominio es mapeada mediante el patrón de **Registro Activo** en la base de datos.
* Las clases de dominio son responsables por guardar su estado, contienen métodos heredados **CRUD** que a su vez representan objetos **DAO** en la comunicación con la base de datos.
* Clases de dominio:

**├── Answer.groovy**

**├── Assessment.groovy**

**├── Category.groovy**

**├── CategoryType.groovy**

**├── CustomField.groovy**

**├── Param.groovy**

**├── Permission.groovy**

**├── Question.groovy**

**├── Role.groovy**

**├── Schedule.groovy**

**├── Survey.groovy**

**└── UserAccount.groovy**

**i18n:** En este directorio se configuran todos los mensajes de la aplicación, cada mensaje se encuentra identificado mediante un código en las vistas de formato GSP. El framework permite utilizar varios lenguajes a mostrar de las vistas, los lenguajes actualizados en la aplicación son español e inglés.

├── messages\_cs\_CZ.properties

├── messages\_da.properties

├── messages\_de.properties

**├── messages\_es.properties**

├── messages\_fr.properties

├── messages\_it.properties

├── messages\_ja.properties

├── messages\_nb.properties

├── messages\_nl.properties

├── messages\_pl.properties

**├── messages.properties**

├── messages\_pt\_BR.properties

├── messages\_pt\_PT.properties

├── messages\_ru.properties

├── messages\_sv.properties

├── messages\_th.properties

└── messages\_zh\_CN.properties

**init:** En este directorio se configuran los parámetros de inicio de la aplicación, contiene la rutina de arranque (método main) y cargue de información para bases de datos en blanco.

**├── BootStrap.groovy**

**└── valueme**

**└── Application.groovy**

**services:** En este directorio se configuran las clases que actúan como servicios de aplicación: Un servicio de aplicación consolida varias tareas individuales, se emplea para la gestión general de usuarios, categorías y cálculo de resultados.

**── valueme**

**├── CategoryService.groovy**

**├── ScoreService.groovy**

**├── SurveyService.groovy**

**└── UserAccountService.groovy**

**taglib:** En este directorio se configuran las etiquetas personalizadas de las vistas para facilitar el desarrollo de componentes visuales reutilizables.

**── valueme**

**├── CustomTagLib.groovy**

**└── SecurityTagLib.groovy**

**views:** Este directorio contiene todas las vistas de la aplicación. Se encuentran organizadas en subdirectorios visibles en el código fuente de la aplicación y a través de la URL del navegador web. Las vistas manejan la notación GSP para componentes del framework de grails y Código HTML.

**├── appInfo**

**├── assessment**

**├── category**

**├── categoryType**

**├── customField**

**├── emails**

**├── layouts**

**├── login**

**├── meci**

**├── param**

**├── permission**

**├── progress**

**├── question**

**├── role**

**├── schedule**

**├── score**

**├── survey**

**└── userAccount**

Cada vista contiene los métodos **CRUD** de las operaciones definidas en el mapa de navegación ej: **assessment**

**/valueme/grails-app/views/assessment/**

**├── create.gsp**

**├── edit.gsp**

**├── index.gsp**

**├── show.gsp**

**├── \_step.gsp**

**├── \_stepShow.gsp**

**└── thankyou.gsp**

**\_step.gsp,\_stepShow.gsp,thankyou.gsp** Son componentes visuales de apoyo para las vistas principales **CRUD**. Manejan un sistema de plantillas para las páginas GSP con funcionalidades específicas.