**Revisión Histórica**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Estado** | **Autor** |
| 1.0 | 10/06/17 | Creación | Kevin Castañeda Zorrilla |

**SDD – Control de disponibilidad de docentes**

**LIBRERÍA PRINCIPAL:**

El proyecto será desarrollado con una arquitectura RESTful, con java desarrollaremos el API que proveerá todos los recursos necesarios que el proyecto cliente lo requiera.

En la capa de persistencia se usó Hibernate, Jackson y gson, herramientas que nos facilitan la manipulación de los datos. Todas las dependencias de este proyecto las encontraremos en el archivo pom.xml.

Para el desarrollo del proyecto cliente usamos el workflow de YEOMAN, esta herramienta nos provee una arquitectura base para el framework AngularJS 1.8. En este proyecto encontraremos dos archivos que nos proporcionan una lista específica de todas las dependencias que se usaron, estos son bower.json y package.json.

**RESPONSABLES:**

La responsabilidad recae en el Jefe de desarrollo y los programadores, quienes son que van a interactuar en directo con las mismas.

**ACTIVIDADES:**

Las actividades principales en cuanto a la gestión de las gemas, son sobre todo el control de las versiones, puesto que las gemas (como cualquier librería), tienden a actualizarse cada cierto tiempo, ya sea por errores encontrados o por nuevas funcionalidades que se desea tener.

Las actividades principales son con respecto al despliegue de los proyectos, en el proyecto en java se deberá generar un archivo “.war” y subirlo al servidor, después de hacer algún cambio, solución de errores o mejoras según los requerimientos. Para el proyecto cliente se generará una carpeta donde están minificados los archivos javascript, css, html y los assets.

Otra actividad principal es el control de versiones, para esté proyecto usamos archivos de configuración los cuales tienen las listas de todas las dependencias y variables globales, en cada entorno de desarrollo y/o producción solo se cambiaran los archivos de variables de configuración, estos archivos no están agregados en el control de versiones ya que por cada entorno pueden variar.

Los roles y el tipo de acceso a cada rol son presentadas en la Tabla 1.

|  |  |
| --- | --- |
| **ROL** | **TIPO DE ACCESO** |
| Gerente del proyecto | * Supervisar * Leer |
| Jefe de desarrollo | * Leer * Supervisar * Escribir * Ejecutar |
| Desarrolladores | * Leer * Ejecutar * Escribir |
| Arquitecto de Software | * Supervisar |

Tabla 1. Roles y Tipos de Acceso

**LIBRERÍAS CONTROLADAS:**

A continuación listamos algunas librerías que utilizaremos en el desarrollo del proyecto, que serán de gran ayuda en el despliegue de funcionalidades:

* **Hibernate:** Hibernate es una herramienta de [mapeo objeto-relacional](https://es.wikipedia.org/wiki/Mapeo_objeto-relacional) (ORM) para la [plataforma Java](https://es.wikipedia.org/wiki/Plataforma_Java) que facilita el mapeo de atributos entre una [base de datos](https://es.wikipedia.org/wiki/Base_de_datos) relacional tradicional y el modelo de [objetos](https://es.wikipedia.org/wiki/Objetos_(programaci%C3%B3n_orientada_a_objetos)) de una aplicación, mediante archivos declarativos ([XML](https://es.wikipedia.org/wiki/XML)) o anotaciones en los [beans](https://es.wikipedia.org/wiki/JavaBean) de las entidades que permiten establecer estas relaciones.
* **Yeoman:** esta herramienta nos ayuda a iniciar la arquitectura de un proyecto prescribiendo mejores prácticas y herramientas para ayudar a mantener un entorno productivo.
* **Schema Plus (schema\_plus):** Es una Gema que nos facilitará la gestión de llaves foráneas e índices en nuestra base de datos física. Dando paso a una gestión más fácil en la relación de modelos (base de datos a nivel lógico).
* **AngularJS:** framework de javascript de código abierto, mantenido por Google, lo utilizamos para crear y mantener nuestra aplicación de una sola página.
* **Google/json:** es una librería para java que nos ayuda a convertir los objetos de java a un formato JSON y viceversa