|  |
| --- |
|  |
|

|  |  |
| --- | --- |
| 1 | Diego Linares |
| 2 | Kenzo Van Waerebeek |
| 3 | Francesca Haro |
| 4 | Carlo Aguilar |

Diagrama de Arquitectura

Para nuestra arquitectura planeamos utilizar las siguientes tecnologías:

*Spring Framework:* es un framework orientado a java utilizado por muchos desarrolladores actualmente. Este tiene un **enfoque en calidad, rapidez de la solución de software y en el orden y coherencia del código implementado**. Siendo la tecnología mas general vista en el curso, Spring framework tiene muchas ventajas, entre ellas: la utilidad de **Spring MVC, donde se nos ofrece facilidad en la disponibilidad del modelo, se nos brindan etiquetas para las peticiones del mapeado del modelo y hallamos la posibilidad de exponer servicios REST (con herramientas como Jersey, Spring HATEOAS, e.o); Spring secutiry, para asegurar las URLs de nuestra web y brinda métodos de autenticación y autorización**; Spring ORM, para representar las tablas de nuestras BD en objetos y viceversa; Spring Batch para manejar el tiempo de respuesta de peticiones en el servidor de nuestra aplicación; Spring integration, para manejar recursos externos o sistemas de trabajo conjunto con nuestra aplicación; testeo y muchas otras más.

*SpringBoot:* Es una solución para la creación de aplicaciones tomando en cuenta bases en Spring de manera **fácil, rápida, independiente y con características prácticamente listas para un despliegue a futuro.** Esta nos permite **evitar configuraciones** y mejorar la experiencia de desarrollo con varios componentes Spring. Asimismo, actualiza las compatibilidades de dependencias especificadas en nuestro proyecto y nos da **soporte de servidores** para realizar el despliegue rápidamente. Para nuestro proyecto permitirá todos los puntos anteriormente mencionados en el menor tiempo posible.

*Spring Data JPA: Spring data es* un proyecto Spring para simplificar la persistencia de datos en repositorios de información. Data JPA es parte de esta familia y **facilita** en nuestro proyecto la **implementación de repositorios basados en JPA**. Esto logra que sea más fácil construir aplicaciones Spring que usen acceso a datos consistentemente.

*Spring Security:* Es un framework de autenticación y control de acceso que nos servirá en el proyecto para **manejar las acciones de los involucrados en el uso de la aplicación final**. Por otro lado, en un futuro podría ayudarnos en temas de seguridad más avanzados como   
protección contra ataques, la fijación de sesiones, clickjacking, falsificación de solicitudes, etc.

*REST:* es una tecnología para crear comunicación entre servidores y clientes de manera eficaz. Una Arquitectura de software vinculada a http y dirigida a escalabilidad de aplicaciones. Todos los datos trabajaran como recursos, y se usaran métodos http estándar definidos (GET, POST, PUT, DELETE, ETC). Nos permitirá múltiples representaciones para pedidos de diferentes servicios o servidores con diferentes aplicaciones. Usa comunicaciones sin estado o stateless (donde no se mantiene persistencia) con cada petición independiente. **Se enfoca en visibilidad, escalabilidad y rendimiento**. Gracias a rest es **fácil dividir microservicios de otros aplicativos**.

*Thymeleaf:* es un motor de plantillas de Java moderno. El objetivo principal es **llevar plantillasnaturales elegantes a su flujo de desarrollo y crear vistas** que se puedan mostrar correctamente en los navegadores. Spring Framework tiene módulos para thymeleaf que, para nuestro proyecto y otros posibles a futuro, **generan integraciones para conectar nuestras propias funcionalidades y mostrar tanto información/datos estáticos como dinámicos**.

*Por ultimo y brevemente, el gestor de base de datos PostgreSQL.*

