

Proyecto de Trabajo Profesional

Animates

Alumnos:

Bello Camilletti, Nicolás (86676) – nbellocamilletti@gmail.com

Biasotti, Pablo (84371) – pablobiasotti@gmail.com

Tutor

Lic. Cosso, Pablo

ABSTRACT

Sitio web para la creación colaborativa de animaciones

Índice

Título	4
Autores.....	4
Tutor	4
Objetivo	4
Introducción.....	4
Alcance.....	5
Requerimientos Funcionales.....	5
Requerimientos no Funcionales.....	8
Hardware	8
Software	8
Seguridad.....	8
Herramientas de Desarrollo.....	9
Hardware.....	9
Dispositivos para desarrollo	9
Software.....	9
Tecnología a utilizar.....	9
Listado de tecnologías.....	10
Metodología.....	11
Cronograma.....	12
Arquitectura.....	18
Componentes	18
Cliente	18
Servidor	18
Persistencia.....	18
Autenticación	19
Sub-proyectos	19

Modelo de Dominio	19
Vista.....	19
Servidor	20
Flujo de operaciones de desarrollo	20
Estadísticas.....	21
Relacionadas con el control de versionado	21
Posibles mejoras.....	22
Conclusiones	23
Referencias	24
Bibliografía	24
Metodología.....	24
Herramientas	24
Plan de Cursado	26
Currículos Autores	26

Título

AniMates - Sitio web para la creación colaborativa de animaciones.

Autores

- Bello Camilletti, Nicolás (86676) - nbellocamilletti@gmail.com
- Biasotti, Pablo (84371) - pablobiasotti@gmail.com

Tutor

- Cosso, Pablo

Objetivo

AniMates permite que varios usuarios puedan crear, reproducir y editar animaciones en un sitio web de modo simultáneo y colaborativo, mejorando la interacción grupal. Asimismo, brinda la posibilidad de descargar las presentaciones para su reproducción sin necesidad de conexión a internet.

Introducción

El siguiente documento está compuesto por dos secciones separando la entrega inicial con la planificación del proyecto de la segunda sección en donde se desarrollan las decisiones de diseño y las conclusiones a las que se llegó luego de realizado el mismo.

Primer sección: Planificación del proyecto

Alcance

Requerimientos Funcionales

- **Registro de usuario y administración de perfil**

El usuario podrá crear una cuenta en el sitio ingresando sus datos personales: Nombre, Apellido, Alias, Mail, Fecha de nacimiento y contraseña. Los mismos podrán ser modificados en cualquier momento desde una sección de perfil de usuario.

- **Autenticación de usuario**

La autenticación de los usuarios podrá realizarse a través de una cuenta registrada o de cuentas de servicios de autenticación de terceros como los provistos por Twitter y Facebook. Cuando un usuario se autentique por primera vez al sitio a través de un servicio de tercero de ser posible se extraerán de la misma los datos necesarios para completar el perfil requiriendo al usuario que ingrese manualmente aquellos campos obligatorios que el servicio no brinde.

- **ABM de proyectos de presentación**

El usuario podrá crear nuevos proyectos de presentaciones, renombrar los ya existentes o incluso eliminar proyectos de su propiedad.

- **Administración del grupo de trabajo sobre los proyectos**

Cada propietario de un proyecto de presentación podrá compartirlo con otros usuarios asignándoles los roles: propietario, visor y editor. El rol visor no podrá realizar ningún cambio sobre la presentación y sólo podrá reproducirla. El rol editor podrá modificar cualquier aspecto de la presentación como el propietario salvando la posibilidad de eliminarla. El propietario es el único que poseerá los permisos necesarios para eliminar una presentación. Así mismo, un propietario, puede ceder la propiedad a otro usuario convirtiéndolo en el nuevo propietario de la presentación.

- **Inserción/Edición de objeto en una presentación**

El usuario podrá insertar o modificar dentro de la presentación objetos de distintos tipos: textos, imágenes y sonidos definiendo su aparición y desaparición dentro de la presentación. Para los objetos que tengan una representación visual, se podrá definir su posición vertical y horizontal dentro de la presentación mediante acciones drag&drop sobre los bordes o el cuerpo del objeto.

- **Animaciones de un objeto**

A cada objeto que el usuario agregue a la presentación se le podrán agregar animación tales como traslación, rotación, redimensión y hacer aparecer/desaparecer gradualmente, entre otros. Dichos efectos se mostrarán en una barra de herramientas podrán ser aplicados seleccionándolos y arrastrándolos sobre los distintos objetos de la presentación.

- **Configuración de una animación**

Cada animación agregada a un objeto poseerá parámetros característicos de la misma que podrán ser configurados individualmente. Dichos parámetros serán mostrados en un cuadro de diálogo al hacer doble click sobre las representaciones en pantalla de las animaciones. El usuario podrá ir pre visualizando las animaciones a medida que las modifica en el mismo diálogo presionado un botón de pre visualización.

- **Línea de tiempo de objetos**

Cada objeto poseerá una línea de tiempo visual donde se podrá ver el momento exacto de inicio y fin de cada animación aplicada al mismo así como también su tiempo de aparición y desaparición en la escena. Todas las líneas de tiempo estarán contenidas dentro de una barra de navegación que podrá ser navegada de forma horizontal en función del tiempo y de forma vertical permitiendo visualizar los distintos objetos. A su vez, esta línea de tiempo, permitirá al usuario visualizar la duración total de la presentación en su conjunto.

- **Pre visualización de un momento específico de la presentación**

Mediante la línea de tiempo el usuario podrá seleccionar un punto específico en el tiempo para pre visualizar el estado de la presentación en ese preciso momento. Esto podrá realizarlo arrastrando un pequeño marcador transversal a todas las líneas de tiempo.

- **Auto guardado de cambio de la presentación**

Cada cambio realizado por los usuarios será guardado de forma automática siempre y cuando exista una conexión a internet activa.

- **Historial de cambios de una presentación**

Cada cambio realizado por los usuarios será registrado y podrá visualizarse un listado de los mismos en un panel denominado “Historial”. Mediante este historial cada usuario podrá deshacer sus cambios utilizando la combinación de teclas Ctrl+Z sobre el cambio seleccionado.

- **Edición colaborativa con otros usuarios**

Los usuarios que tengan permisos de edición podrán realizar cambios en forma simultánea con otros usuarios que estén visualizando o editando la presentación sin la necesidad de bloquear el acceso al proyecto de presentación.

- **Notificaciones de errores**

Los usuarios serán notificado con mensajes en pantalla de los errores ocurridos como puede ser la desconexión a internet, o el borrado de la presentación mientras está siendo editada, entre otros.

- **Visualización de una presentación**

Todos los usuarios con acceso a un proyecto podrán visualizar la presentación mediante un botón que ejecutará la reproducción de inicio a fin con la posibilidad de pausar y reanudar.

- **Descarga/exportación de la presentación**

Se permitirá descargar la presentación en un formato que permita reproducirla de manera offline, así como integrarla con sitios web existentes, descargando los archivos necesarios (Imágenes, código JavaScript, hojas de estilo, etc.).

Requerimientos no Funcionales

Hardware

El servidor de la aplicación se desarrollará para ser instalado en uno de los servicios de cloud computing por ej. Windows Azure Websites [24], Nodejitsu [25] o Heroku [26] por lo que no requerirá un hardware específico fuera de lo que este tipo de soluciones ofrecen. Lo que se requerirá es que tengan soporte para Node.js [8].

El acceso a la aplicación será mediante un sitio web por lo que sólo se requerirá un dispositivo con un navegador de internet moderno por ej. Firefox, Chrome, Safari.

Software

- Servidor
 - Sistema operativo que cuente con una implementación de Node.js [8] (open source)
 - Servidor de base de datos relacional PostgreSQL [1,5] (open source)
 - Servidor de base de datos NoSQL [1] MongoDB [1,4] (open source)
- Cliente
 - Navegador Web con soporte de tecnología HTML5 tipo Chrome o Firefox (open source)

Seguridad

Para mantener un contexto de privacidad dentro del sitio se contará con autenticación y autorización de usuarios, permitiendo autenticarse utilizando servicios de terceros o una cuenta propia del sistema. Esta identificación en conjunto con el mecanismo para compartir presentaciones restringirá el acceso a documentos privados a menos que el usuario decida lo contrario y cambie los permisos sobre sus presentaciones.

En lo referido a las comunicaciones entre servidor web y navegador se aplicará el uso del protocolo SSL/TLS a todas aquellas que se consideren sensibles (ej. autenticación, modificación de contraseña, etc.) dicho protocolo mantiene la comunicación cifrada de forma transparente para el usuario.

Para todos aquellos datos también sensibles que necesiten ser persistidos en un soporte físico como puede ser una base de datos o un archivo, se realizará también una cifrado mediante el algoritmo simétrico AES o similar.

Herramientas de Desarrollo

Hardware

Dispositivos para desarrollo

- MacBook Air (Mid 2012)
 - Procesador: 1.8GHz Intel Dual-Core Core i5, Turbo Boost up to 2.8GHz
 - Memoria RAM: 8GB 1600MHz DDR3L SDRAM
 - Disco Rígido: 256GB Flash Storage
 - SO de 64 bits
 - OS X Mountain Lion v10.8.5
- MacBook Pro (Mid 2009)
 - Procesador: 2.53 GHz Intel Core 2 Duo
 - Memoria RAM: 4GB 1067 MHz DDR3
 - Disco Rígido: 250GB
 - SO de 64 bits
 - OS X Snow Leopard v10.6.8

Software

Tecnología a utilizar

Se eligió basarse en las tecnologías web de última generación como lo son HTML 5, CSS 3 y JavaScript [2,6] para lo que es el cliente web, Node.js [8] para el servidor y websockets para la comunicación entre ambos.

Por el lado del cliente, se eligió trabajar con estas tecnologías dado que nos permiten lograr nuestro objetivo basándonos en estándares gratuitos eliminando la dependencia de implementaciones de software de terceros como puede ser Flash o Silverlight.

Por el lado del servidor, Node.js [8] nos posibilita la creación de un servidor web en el mismo lenguaje que se utiliza del lado del cliente, y nos da la ventaja para realizar un proyecto que busca la colaboratividad entre los usuarios con el sistema de eventos y la facilidad para agregar soporte para websockets los cuales nos permiten generar distintos canales de comunicación Full-Duplex sobre una única conexión TCP entre el browser y el servidor.

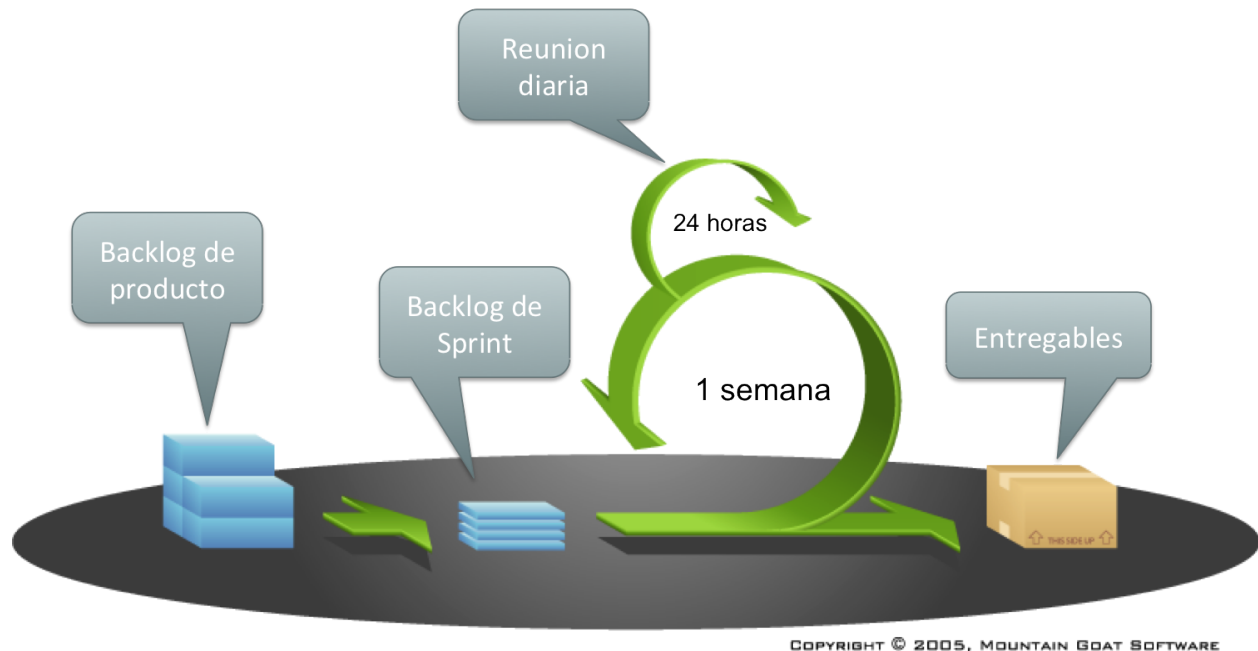
Finalmente, como todo sitio web pretende lograr una gran demanda del servicio en el futuro, se diseñará el sistema para escalar de manera flexible. Para ello se utilizará una base de datos NoSQL [1] orientada a documentos (MongoDB [1,4]) las cuales permiten el manejo de grandes volúmenes de datos de forma eficiente y la flexibilidad necesaria para la escalabilidad de nuestro proyecto, teniendo en cuenta además que las estructuras de datos que se esperan manejar no son soportadas por base de datos relacional. Al ser necesaria también una escalabilidad a nivel de procesamiento se utilizará como infraestructura la denominada nube (por ejemplo, Nodejitsu [25], Windows Azure [24], Heroku [26], etc), las cuales permiten el manejo de infraestructura por medio de virtualización de manera flexible pero transparente para el sistema, permitiendo escalar en función de las necesidades o requerimientos acordes al uso/tráfico del sitio.

Listado de tecnologías

- Bases de datos
 - MongoDB (NoSQL - orientada a documentos) [1,4]
 - PostgreSQL (relacional) [1,5]
- Lenguaje
 - JavaScript [2,6]
- IDE
 - Sublime text 2 / editor de texto [7]
- Framework
 - Server-side
 - Node.js [8]
 - Express [9]
 - Mocha [10]
 - Passport [11]
 - Jade [12]
 - Socket.io [13]
 - Client-side
 - JQuery [14]
 - JQueryUI [15]
 - Twitter Bootstrap [16]
 - Raphael.js [17]
 - Fabric.js [18]
- Control de versiones
 - Git [19]
 - GitHub [20]/Bitbucket [21]
- Documentos
 - Google Docs [22]
- Sistema de tickets, seguimiento del proyecto
 - RedMine [23]
- Hosting - Cloud Computing solution
 - Windows Azure Websites [24]
 - Nodejitsu [25]
 - Heroku [26]
- Build server
 - Jenkins [27]
 - Grunt [28]

Metodología

Durante el transcurso del proyecto se empleará el framework ágil de desarrollo SCRUM [3]. Este, es un proceso ágil que nos permite centrarnos en ofrecer el más alto valor de negocio en el menor tiempo. El proyecto avanza en una serie de Sprints (iteraciones) cuya duración está limitada a una semana. La razón por la cual se elige una iteración de esta duración es porque nos permitirá ajustar las planificaciones más rápidamente. El producto es diseñado, codificado y probado en este período.



Cada 3 iteraciones se organizará una reunión en la que participarán los desarrolladores y el Product Owner donde se evaluará el trabajo realizado.

Para el seguimiento del trabajo se utilizará Redmine [23] que es una herramienta para la administración de proyectos.

Cronograma

El proyecto requiere un esfuerzo total de **1125 horas/hombre**. Esta estimación se realizó sobre una base de 707 horas/hombre estimadas para el desarrollo, agregando un 25% de esfuerzo de pruebas por cada tarea (209 horas/hombre), otro 25% de esfuerzo dedicado a tareas de administración y a amortiguadores de estimación (209 horas/hombre). Contando con dos personas dentro del equipo las cuales pueden dedicar entre 26 y 36 horas semanales y ordenando las historias de usuario de acuerdo al valor que le entregan a un usuario final se decidió utilizar la siguiente planificación de iteraciones semanales:

Titulo historia	Horas	Iteración
Análisis funcional y descripción de requerimientos de alto nivel	24	1
Definición de requerimientos de bajo nivel y estimación	17	2
Planificación	10	
Armado de entregables de plan de proyecto	6	
Configuración de entornos de desarrollo	10	3
Configuración de repositorio de control de versiones	4	
Creación del proyecto con las dependencias	16	
Creación del proyecto de integración continua	16	4
Creación de la estructura del sitio web	6	
Modelo de dominio - Diseño	16	5
Modelo de dominio - Implementación	16	
Capa de persistencia - Investigación de uso de MongoDB y NodeJS	10	6
Capa de persistencia - Diseño de capa de persistencia	6	
Capa de persistencia - Implementación con MongoDB	10	
Modelo de dominio - Persistencia	24	7
Creación de estructura MVC en el servidor	7	
Creación de estructura MVC desde JS (cliente)	10	8
Definición de estética básica del sitio (Colores, tipos y tamaños de fuente)	7	
Creación de página básica con mecanismos comunes a todas las páginas	6	
Control JS - Canvas - Previsualización de un momento de la presentación	18	9

Control JS - TimelineBrowser - Selección de un momento específico en el tiempo	6	
Control JS - PresentationEditor - Integración Canvas y TimelineBrowser	7	
Pantalla de edición - Sección Header/Toolbar - Mockup UX	6	10
Pantalla de edición - Sección Header/Toolbar - Maquetado HTML+CSS	10	
Pantalla de edición - Sección Canvas - Mockup UX	10	
Pantalla de edición - Sección Canvas - Maquetado HTML+CSS	10	
Pantalla de edición - Sección TimelineBrowser - Mockup UX	10	11
Pantalla de edición - Sección TimelineBrowser - Maquetado HTML+CSS	10	
Investigación de Tecnologías para renderizado	16	
Control JS - PresentationEditor - Layout	10	12
Control JS - Canvas - Layout	6	
Control JS - Canvas - Definición de interfaz de CanvasObject	10	
Control JS - Canvas - CanvasObjects - Layout	6	
Control JS - PresentationEditor - Agregar nuevo Object - Upload File	12	13
Control JS - PresentationEditor - Agregar nuevo Object - Image	5	
Control JS - PresentationEditor - Agregar nuevo Object - Sound	5	
Control JS - PresentationEditor - Agregar nuevo Object - Text	4	
Control JS - Canvas - Agregar Objeto	7	
Control JS - Canvas - CanvasObjects Events - Click handler	4	14
Control JS - Canvas - CanvasObjects Events - Move handler	5	
Control JS - Canvas - CanvasObjects Events - Resize handler	5	
Control JS - Canvas - CanvasObjects Events - Delete handler	4	
Control JS - Canvas - TextCanvasObject - Layout	4	
Control JS - Canvas - ImageCanvasObject - Layout	4	
Control JS - PresentationEditor - Definición de interfaz de EffectObject	7	
Pantalla de edición - Diseño y Implementación - Visualización de efectos de forma general	4	15
Control JS - PresentationEditor - Implementación de interfaz de EffectObject	7	

Pantalla de edición - Diseño y Implementación - Visualización de FadeOut	1	
Control JS - PresentationEditor - Implementación de FadeOutEffectObject	10	
Pantalla de edición - Diseño y Implementación - Visualización de FadeIn	1	
Control JS - PresentationEditor - Implementación de FadeInEffectObject	10	
Pantalla de edición - Diseño y Implementación - Visualización de Move	11	16
Control JS - PresentationEditor - Implementación de MoveEffectObject	10	
Pantalla de edición - Diseño y Implementación - Visualización de Rotate	4	
Control JS - PresentationEditor - Implementación de RotateEffectObject	10	
Pantalla de edición - Diseño y Implementación - Visualización de Resize	4	17
Control JS - PresentationEditor - Implementación de ResizeEffectObject	10	
Pantalla de edición - Listado de Efectos - Mockup UX	6	
Pantalla de edición - Listado de Efectos - Maquetado HTML+CSS	10	
Pantalla de edición - Agregar nuevo efecto a un objeto	6	18
Pantalla de edición - Agregar nuevo efecto a un objeto - Representación en timeline	5	
Control JS - PersentationEditor - Edición visual de duración de un efecto en timeline	5	
Control JS - PersentationEditor - Diseño de UX de parámetros específicos de una animación	10	
Control JS - PersentationEditor - Persistencia del nuevo efecto	4	19
Control JS - TimeLine - Desplazamiento temporal	16	
Control JS - TimeLineBrowser - Desplazamiento temporal	10	
Control JS - TimeLineBrowser - Desplazamiento por objetos	6	20
Control JS - TimeLineBrowser - Layout	10	
Control JS - TimeLineBrowser - Agregar Objeto	6	
Control JS - TimeLine - Layout	6	
Control JS - TimeLine - TimeLapse layout	5	
Control JS - TimeLine - TimeLapse Events - Click raise	4	
Control JS - TimeLine - TimeLapse Events - Move raise	5	

Control JS - TimeLine - TimeLapse Events - Resize (Reduce/Enlarge) raise	5	21
Control JS - Object Timeline - Click handler	4	
Control JS - Object Timeline - Move handler	5	
Control JS - Object Timeline - Resize handler	5	
Pantalla de registro de nuevo usuario - Mockup UX	4	
Pantalla de registro de nuevo usuario - Maquetado HTML+CSS	6	
Pantalla de registro de nuevo usuario - Creación de nuevo usuario	6	
Pantalla de registro de nuevo usuario - Investigación y modo de uso de Passport	5	22
Pantalla de registro de nuevo usuario - Conexión con autenticación de terceros (UI)	6	
Pantalla de perfil de usuario - Mockup UX	6	
Pantalla de perfil de usuario - Maquetado HTML+CSS	10	
Pantalla de perfil de usuario - Actualización de perfil de usuario	7	
Guardado de settings del usuario	10	23
Pantalla de login - Mockup UX	4	
Pantalla de login - Maquetado HTML+CSS	6	
Pantalla de login - Autenticación de usuarios	6	
Pantalla de login - Login encriptado	5	
Configuración de plugin de autenticación	6	24
Pantalla de login - Integración con usuarios registrados	5	
Pantalla de login - Conexión con autenticación de terceros	4	
Pantalla principal de usuario - Mockup UX	10	
Pantalla principal de usuario - Maquetado HTML+CSS	16	25
Pantalla principal de usuario - Creación de un proyecto nuevo	6	
Pantalla de listado de proyectos - Mockup UX	6	
Pantalla de listado de proyectos - Maquetado HTML+CSS	12	26
Pantalla de listado de proyectos - Integración con los proyectos del usuario	6	
Pantalla de listado de proyectos - Eliminación de un proyecto	4	

Pantalla de listado de proyectos - Edición de un proyecto (Atributos estáticos: nombre, descripción, etc.)	6	
Pantalla de compartir - Mockup UX	4	
Pantalla de compartir - Maquetado HTML+CSS	6	27
Diseño del mecanismo para compartir	4	
Pantalla de compartir - Compartir con otro usuario	5	
Pantalla de compartir - Autocomplete de nombres de usuario	4	
Pantalla de compartir - Asignación de roles	5	
Pantalla de compartir - Cesión de propiedad	5	
Corrección de errores 1	23	28
Pantalla de historial de cambios - Mockup UX	5	
Pantalla de historial de cambios - Maquetado HTML+CSS	6	
Librería de manejos de cambios - Diseño de modelo de cambios (Que objeto, que modificación, propiedades)	10	29
Librería de manejos de cambios - Implementación de modelo de cambios (Que objeto, que modificación, propiedades)	12	
Librería de manejos de cambios - Comunicación Browser-Server	12	
Librería de manejos de cambios - Persistencia	12	30
Librería de manejos de cambios - Revertir a una versión	7	
Pantalla de edición - Mockup UX	11	
Pantalla de edición - Maquetado HTML+CSS	12	31
Librería de edición en tiempo real - Mutex y Encolado de modificaciones sobre un mismo objeto	12	
Librería de edición en tiempo real - Comunicación Browser-Server (Replicación de cambios)	16	32
Librería de edición en tiempo real - Integración con librería de manejos de cambios	16	
Pantalla de historial de cambios - Actualización ante cambios de otros usuarios	6	33
Pantalla de edición - Actualización ante cambios de otros usuarios	6	
Pantalla de visualización - Mockup UX	7	

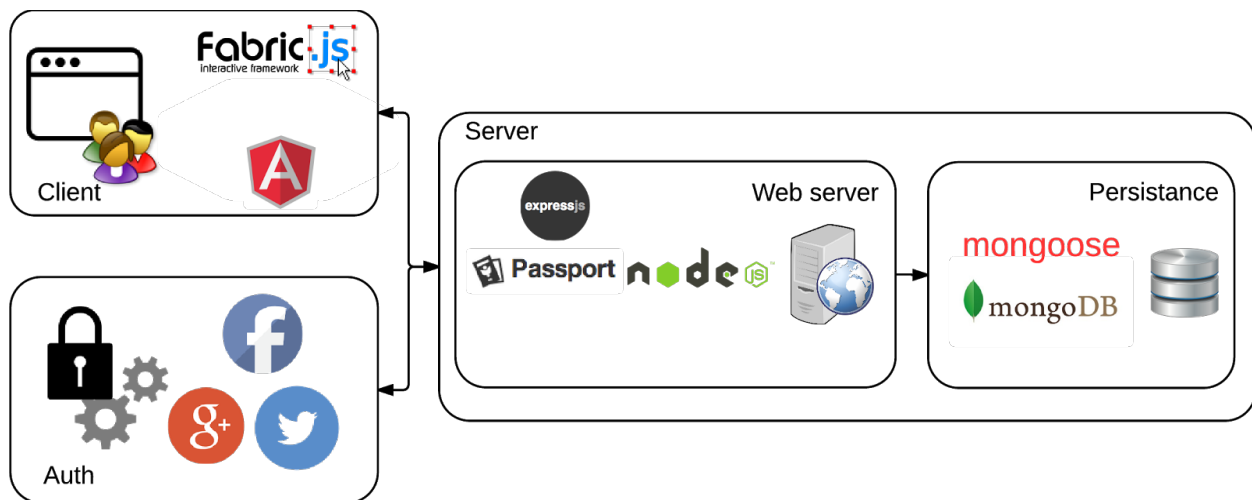
Pantalla de visualización - Maquetado HTML+CSS	7	
Librería de reproducción de la presentación - Reproducción de un efecto	10	
Librería de reproducción de la presentación - Reproducción de una línea de tiempo	10	34
Librería de reproducción de la presentación - Reproducción de varias líneas de tiempo	12	
Pantalla de exportación - Mockup UX	5	
Pantalla de exportación - Instrucciones para visualización	5	
Pantalla de exportación - Maquetado HTML+CSS	5	35
Pantalla de exportación - Empaquetado de archivos necesarios	10	
Pantalla de exportación - Descarga de archivo	1	
Análisis de performance y optimización	16	
Corrección de errores 2	23	36
Creación de presentación introductoria - Español	16	37

Segunda sección: Implementación del proyecto

Arquitectura

Componentes

Dentro del sistema podemos identificar un total de cuatro componentes claves.



Cliente

Desarrollado enteramente con las tecnologías HTML5 y Javascript. Adicionalmente se utilizaron los frameworks **AngularJS** para manejo de elementos DOM e interacción con el usuario y **Fabric.js** para el manejo de Canvas.

Servidor

Montado sobre **node.js** se encuentra el módulo **express** el cual brinda herramientas orientadas a servidores de sitios web permitiendo a través de la aplicación de capas llamadas middlewares el enriquecimiento del objeto request durante su procesamiento con información relevante como pueden ser datos de usuario, sesión, etc.

Persistencia

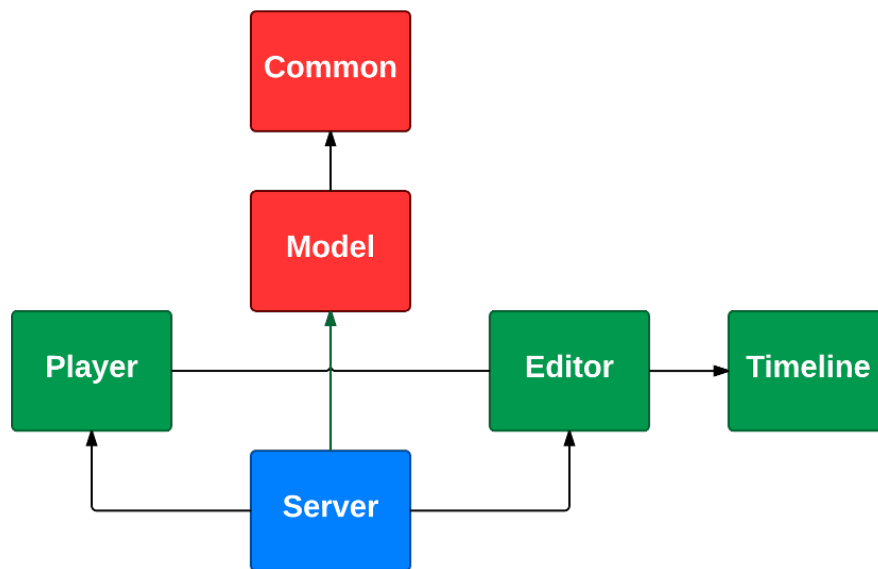
Como motor de base de datos se utiliza **MongoDB** el cual entra en la categoría de las denominadas bases de datos NoSQL orientadas a documentos. Adicionalmente se utilizó el framework **mongoose** que brinda una abstracción que permite modelar los objetos a persistir definiendo esquemas. Los mismos no solo modelan la estructura de los datos sino que también permiten el agregado de comportamiento y validación relacionados a la persistencia.

Autenticación

Utilizando el framework **passport** el sistema permite la autenticación con los servicios de **twitter**, **facebook** y **google+** brindando al usuario la posibilidad de reutilizar sus cuentas. Para aquellos usuarios que no cuenten con una existe la posibilidad de crear un usuario en nuestro sistema. Las peticiones subsiguientes a la autenticación son validadas por medio del protocolo **jwt** (JSON web token).

Sub-proyectos

Para evitar el acoplamiento y favorecer la modularización del sistema el mismo se dividió en 6 proyectos agrupados en 3 categorías:



Modelo de Dominio

Common: librería de utilidades de uso común dentro de las cuales podemos destacar las que nos permiten realizar un enfoque de orientación a objetos con herencia dentro de javascript que es un lenguaje orientado a objetos pero basado en prototipos.

Model: es el modelo de dominio que contiene las reglas de validación relacionadas al negocio.

Vista

Player: permite la reproducción de una animación tanto dentro del sitio como de forma offline.

Editor: permite la edición de animaciones en un navegador web. Ofreciendo una interfaz por medio del cual cualquier módulo externo puede suscribirse a los cambios que se vayan realizando.

Timeline: permite la visualización de elementos en un contexto temporal. Ofrece una interfaz de eventos a los pueden ser capturados por módulos externos para recibir notificaciones de los cambios que el usuario vaya realizando. Cabe destacar que tiene un modelo de representación independiente al modelo de Animates lo que permite que pueda ser reutilizado en otros proyectos externos.

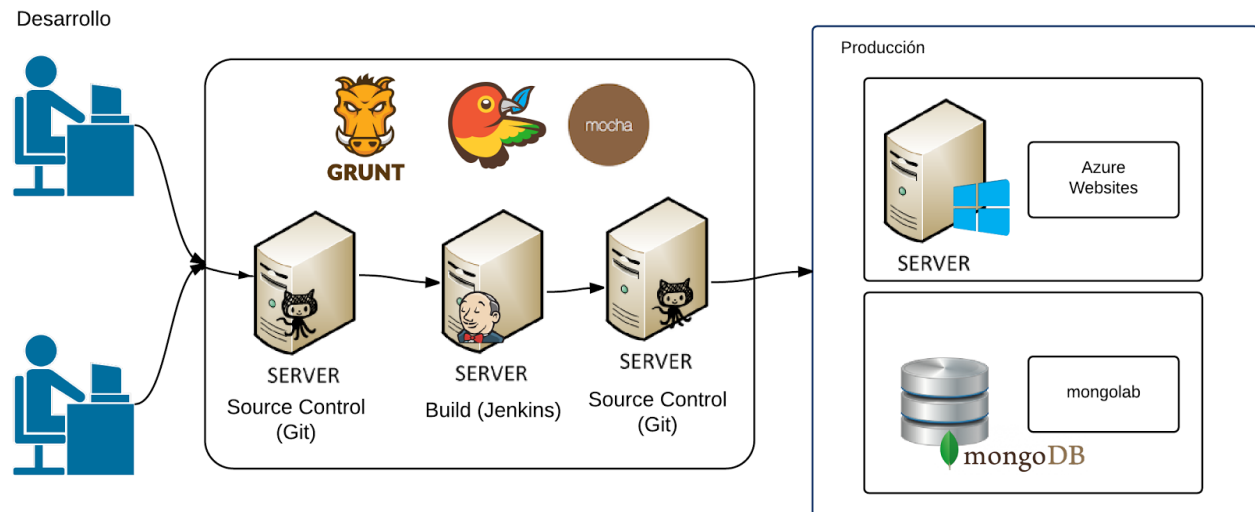
Servidor

Server: agrupa e integra al resto de los proyectos montandolos sobre un servidor web agregando la autenticación de usuarios y persistencia de datos.

Flujo de operaciones de desarrollo

Para el desarrollo del sistema, se utilizó un servidor de integración continua (**Jenkins**), el cual ejecuta un conjunto de scripts creados con **Grunt.js** cada vez que se envían cambios al repositorio alojado en **GitHub**. Dentro de las tareas que se realizan, se destacan la ejecución de los tests unitarios y la ejecución de la herramienta **JSHint**, la cual realiza un control sobre el código escrito para asegurar que se cumplan con ciertos patrones y recomendaciones tomadas de las convenciones de código definidas en especificación ECMAScript. Esto nos permitió asegurarnos el correcto funcionamiento del sistema de forma permanente.

Luego de esta etapa de revisión del código, se realizan tareas para la puesta en producción del mismo, para lo cual se destacan tareas como la de concatenación de archivos de JavaScript y minificación de todos los archivos orientados a la optimización.

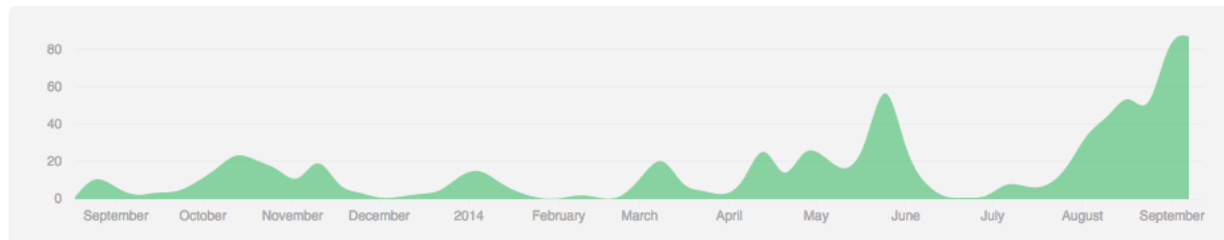


Estadísticas

Relacionadas con el control de versionado

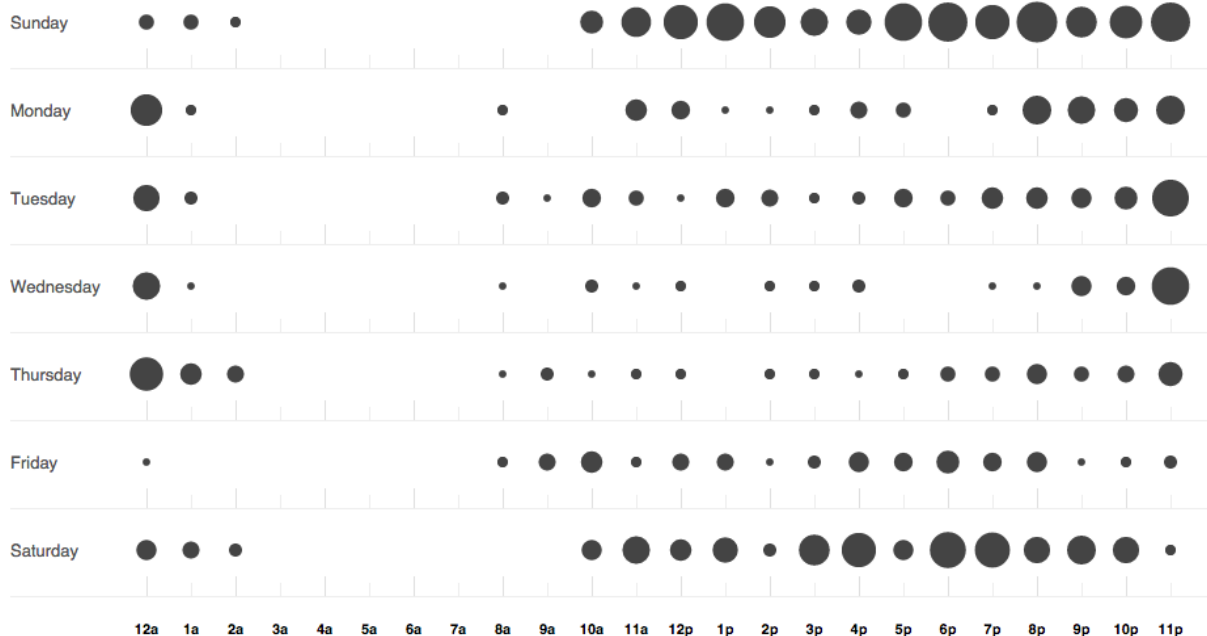
Frecuencia de commits en función al tiempo

Podemos observar que no pudimos mantener una frecuencia estable en el desarrollo del sistema, por motivos laborales y personales.



Frecuencia de commits en función al horario del día

Podemos observar que los fin de semanas fueron los días donde se realizó la mayor parte del trabajo, así como también durante la semana luego del horario laboral.



Posibles mejoras

Se detectaron las siguientes posibles mejoras para realizar

- Agregar soporte para la creación de un archivo de video descargable desde el servidor
- Agregar la creación y visualización de la imagen previa para cada proyecto.
- Agregar soporte para sonido.
- Mejorar el soporte cross-browser incluyendo soporte para Internet Explorer y versiones más antiguas de algunos browsers.
- Agregar la posibilidad de realizar zoom en el canvas.
- Agregar el soporte para la creación de escenas, permitiendo una mejor interacción para animaciones de larga duración.
- Mejorar la visualización de los otros usuarios agregando elementos visuales que permitan ver quienes están conectados en el mismo momento, que estan modificando y quien fue el responsable de cada cambio.
- Mejorar usabilidad del editor agregando soporte para accesos directos utilizando el teclado, entre otros.
- Integración con redes sociales permitiendo compartir animaciones publicas en Facebook, Twitter, etc., integrándose con estos sistemas.

Conclusiones

Consideramos que la elección de las tecnologías fue correcta en función de los requerimientos técnicos del proyecto, no obstante muchas de las herramientas y tecnologías utilizadas al estar surgiendo y encontrarse en fase de desarrollo no contaban con versiones estables y documentación apropiada. Por las mismas razones algunas tareas inherentes al desarrollo como el debugging requirió por momentos esfuerzos muy grandes. Esto ocasionó un retraso importante el proyecto por lo que se decidió recortar las funcionalidades de reproducción de sonido y la posibilidad de poder deshacer y rehacer cambios recientes realizados por el usuario.

El enfoque similar al de un emprendimiento que hemos decidido darle al proyecto nos resultó muy satisfactorio en lo referido a la metodología de trabajo y al resultado obtenido. No obstante hemos podido realizar, por cuestiones de tiempo, controles previos a la puesta en producción orientados a la detección de fallos debidos a los procesos de optimización (ofuscación de código y compresión de recursos).

En lo referido al diseño de la aplicación conocemos de ciertas limitaciones que podrían considerarse como deuda técnica que nos imposibilitaron poder realizar renderizaciones del lado del servidor. Esto hubiera permitido poder crear vistas previas de las animaciones o crear un video a partir de las mismas. Para evitar esta problemática se debería desarrollar un nuevo módulo encargado del renderizado (al momento de finalizar este proyecto dicha responsabilidad está muy acoplada a los componentes asociados a la vista) de forma tal de poder re-utilizarlo tanto en la vista como en el servidor. Por otro lado, si bien hemos realizado una investigación respecto de las opciones que nos brindaban los distintos frameworks de dibujado en canvas y no considerábamos la posibilidad de tener que cambiarlo, hubiera sido acertado realizar una abstracción propia que encapsule a FabricJS de forma de poder reemplazarlo de ser necesario sin afectar al resto de los componentes.

Referencias

Bibliografía

- [1] NoSql
 - Seven Databases in Seven Weeks - Eric Redmond y Jim R. Wilson. Ed. The Pragmatic Bookshelf
 - NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence de Pramod J. Sadalage y Martin Fowler. Ed. Addison-Wesley Professional.
 - MongoDB: The Definitive Guide de Kristina Chodorow y Mike Dirolf. Ed. O'Reilly Media
- [2] JavaScript
 - JavaScript: The Good Parts de Douglas Crockford. Ed. O'Reilly Media
 - JavaScript Patterns de Stoyan Stefanov. Ed. O'Reilly Media
 - Learning JavaScript Design Patterns de Addy Osmani. Ed. O'Reilly Media
 - High Performance JavaScript de Nicholas C. Zakas. Ed. O'Reilly Media
 - JavaScript Web Applications de Alex MacCaw. Ed. O'Reilly Media

Metodología

- [3] Scrum
 - <http://www.mountaingoatsoftware.com/topics/scrum>
 - http://www.scrumalliance.org/pages/what_is_scrum

Herramientas

- Bases de datos
 - [4] MongoDB: <http://www.mongodb.org>
 - [5] PostgreSQL: <http://www.postgresql.org>
- Lenguaje
 - [6] JavaScript: <https://developer.mozilla.org/en-US/docs/JavaScript>
- IDE
 - [7] Sublime text 2: <http://www.sublimetext.com>
- Frameworks
 - Server-side
 - [8] Node.js: <http://nodejs.org/>
 - [9] Express: <http://expressjs.com>
 - [10] Mocha: <http://visionmedia.github.io/mocha/>
 - [11] Passport: <http://passportjs.org>
 - [12] Jade: <http://jade-lang.com>
 - [13] Socket.io: <http://socket.io>
 - Client-side
 - [14] JQuery: <http://www.jquery.com>
 - [15] JQueryUi: <http://www.jqueryui.com>
 - [16] Twitter Bootstrap: <http://twitter.github.com/bootstrap/>
 - [17] Raphael.js: <http://dmitrybaranovskiy.github.io/raphael/>
 - [18] Fabric.js: <http://dmitrybaranovskiy.github.io/fabric.js/>

- Control de versiones
 - [19] Git: <http://www.git-scm.com>
 - [20] GitHub: <https://www.github.com>
 - [21] Bitbucket: <https://www.bitbucket.org>
- Documentos
 - [22] Google Docs: <https://docs.google.com>
- Sistema de tickets, seguimiento del proyecto
 - [23] Redmine: <http://www.redmine.org>
- Hosting - Cloud Computing solution
 - [24] Windows Azure Websites: <http://www.windowsazure.com/>
 - [25] Nodejitsu: <https://www.nodejitsu.com/>
 - [26] Heroku: <http://www.heroku.com>
- Build server
 - [27] Jenkins: <http://www.jenkins-ci.org>
 - [28] Grunt: <http://gruntjs.com>

Plan de Cursado

Tanto el alumno Nicolás Bello Camilletti, como el alumno Pablo Biasotti, no se encuentran cursando ninguna asignatura ni adeudan ningún final de la cursada, siendo este trabajo la última asignatura pendiente para finalizar la carrera.

Currículos Autores



NICOLÁS BELLO CAMILLETTI

Fecha de nacimiento
08/09/1985

Domicilio
Bulnes 620 6to piso departamento 39
Capital Federal
Argentina

Contacto
+54-11-4863-9586
+54-11-3656-0499
nbellocamilletti@gmail.com

EXPERIENCIA

Desarrollador
Southworks S.R.L.
Mayo de 2011 - Actualidad

Desarrollo como partner de Microsoft. Fuerte foco en Windows Azure. Creación de herramientas y ejemplos para Windows Phone. Creación de herramientas y ejemplos para Windows 8. Contacto con clientes por medios escritos y orales. Utilización de metodologías ágiles.

ANTECEDENTES DOCENTES

Colaborador en la materia Seminario de Ingeniería en Informática (Cátedra Pablo Cosso)
Universidad de Buenos Aires - Facultad de Ingeniería
Agosto 2011 - Presente

Clases de Groovy y Grails. Atención de consultas y correcciones. Creación de sitio y grupo correo de la materia

EDUCACIÓN

Ingeniería en informática, Universidad de Buenos Aires
Grado — 2004 - Actualidad (En curso)
Total de materias aprobadas: 40
Promedio: 7,4

Licenciatura en sistemas, Universidad de Buenos Aires
Grado — 2004 - 2012 (Finalizado)
Total de materias aprobadas: 29
Promedio: 7,6

Escuela Superior De Comercio Carlos Pellegrini – U.B.A.
Secundario — 1998 - 2003

CONOCIMIENTOS ADICIONALES

- Idioma Inglés:
 - First Certificate in English, Cambridge University ESOL. Grade B
 - Level 10 - Laboratorio de Idiomas – U.B.A.
- Lenguajes de programación: C#, Groovy, Javascript
- Domain Driven Design, Patrones de diseño
- Programación orientada a objetos y funcional
- Cloud Computing
- Bases de datos SQL y NoSQL
- Programación Web y de dispositivos móviles
- Armado y Dictado de cursos y charlas

OTROS

- 27 de junio de 2012 - Presentación en Microsoft Argentina en el contexto de Windows Azure Workshop sobre los toolkits de Microsoft DPE de Windows Azure para dispositivos móviles y Windows 8.
- Febrero de 2013 - Certificación de Microsoft 70-599: "Pro: Designing and Developing Windows Phone Applications"



PABLO BIASOTTI

Fecha de nacimiento
07/01/1984

Domicilio
Amenabar 879
Capital Federal
Argentina

Contacto
+54-11-4025-7334
+54-11-4784-0372
pablobiasotti@gmail.com

EXPERIENCIA

Analista Programador Senior / Team Leader Barcelona/04 Computing Group

Junio de 20013-Presente

Líder de equipo de mantenimiento, soporte y desarrollos de mejoras a corto plazo de 7 proyectos relacionados a la monitorización de negocios e infraestructura los cuales que integran diferentes tecnologías como Asp.Net, C#, C++ y Python.

Analista Programador Senior Barcelona/04 Computing Group

Junio de 2007-Presente

Desarrollo de aplicaciones web orientadas a la visualización de información generada por sistemas de monitorización de infraestructura y negocio. Desarrollos asociados a monitorización de base de datos MS SQL Server 2005/2008

Analista Programador Semi Senior Sistemas Bejerman

Mayo de 2005 - Diciembre de 2006

Mantenimiento y desarrollo de nuevas funcionalidades del módulo de producción del sistema E-Flexware. VB6, MS SQL 2000 / Access 2000, Crystal Reports.

Analista Programador Junior Seincomp Informática S.R.L.

Julio de 2004 - Abril de 2005

Programador en VB6, MS SQL Server, MS Access 2000/97, Crystal Reports, Active Reports. Conocimientos de Bases DBASE de Tango.

Analista Programador Junior Buscadordecoracion.com

Enero de 2004 - Julio de 2006

Desarrollo y mantenimiento del sitio (ASP 3.0, MS Access 2000, HTML, Javascript, css, etc). Soporte Técnico de red y Pcs. Tareas varias referentes a administración de hosting, registro de dominios, etc.

EDUCACIÓN

Ingeniería en informática, Universidad de Buenos Aires

Grado — 2004 - Presente
Promedio: 6,475

Licenciatura en sistemas, Universidad de Buenos Aires

Grado — 2004 - 2012
Promedio: 6,689

CONOCIMIENTOS ADICIONALES

- Idioma Inglés: Avanzado Oral y escrito

