Instituto Politécnico Nacional

Escuela Superior de Cómputo

SOFTECH

Ingeniería de Software

Prof. Marco Antonio Dorantes González

García Sánchez Alexis Andrés

Dávila Méndez Juan Manuel

Quintana Ruiz Ajitzi Ricardo

Zepeda Flores Alejandro de Jesús

Ciudad de México, 31 de Mayo de 2019

# Índice

[**Índice**](#_4u3az3hyh024) **2**

[**Introducción**](#_v1s21fmtlgrk) **5**

[**Marco teórico**](#_muw7r014m6oo) **6**

[Industria 4.0](#_i8gaqnljjagf) 7

[Retos de la industria 4.0](#_eq9syfnml0ju) 7

[**Estado del arte**](#_5lufa4k0bi2t) **11**

[**Métricas de estimación**](#_id2wxaz154wi) **12**

[Puntos de función](#_dd8fjqa6rorv) 12

[**Cronograma**](#_8fmoe0uf1d80) **14**

[**Metodología**](#_s01uuu5lu26) **15**

[Metodología SCRUM](#_qqvdxadjlrdk) 15

[Roles de Scrum](#_auzdktapczsq) 15

[Fases de la metodología Scrum](#_inradnv8d78f) 16

[Beneficios](#_budi65lml78c) 16

[**Modelo de procesos**](#_4xg7zcid1fqv) **17**

[Basado en prototipos](#_byiskehycqjv) 17

[Etapas](#_wkl9yhdted2a) 17

[Ventajas](#_vk9rmu42o6nh) 17

[**Modelo de negocios**](#_567x7gr6mmvy) **18**

[Long Tail](#_zbezahhdciz9) 18

[**Técnicas de recolección de datos**](#_wirpxnih1hji) **20**

[**Requerimientos**](#_5j43p9v2gg6n) **22**

[**Clave**](#_de9d46k98lzs) **22**

[**Descripción**](#_de9d46k98lzs) **22**

[**Requerimientos Funcionales**](#_it3wq5hlzoef) **23**

[**Requerimientos no Funcionales**](#_69y0fvue8dtk) **24**

[Diagrama de casos de uso](#_wr6xhh2ts9j6) 25

[Diagrama de secuencias](#_60pc9vy3a928) 26

[Diagrama de clases](#_b5syub5j0lp4) 30

[Diagrama de actividades](#_60pc9vy3a928) 31

[**Pruebas**](#_yi8r2g19uvtf) **33**

[Prueba de integración](#_7j494xy8cd72) 33

[Mantenibilidad](#_gz9l31dlmfn6) 33

[Pruebas de estrés](#_a01y85iju75q) 34

[Pruebas de portabilidad](#_ymx8rubd2kw5) 34

[**Tecnologías a utilizar**](#_gb4ql7vp4hdy) **35**

[**Modelo de entidades del negocio**](#_nyb1tkbffsnu) **36**

[Descripción de entidades del negocio](#_o2h5xwognki1) 36

[Atributos de ‘Usuario’](#_bwliye3bwsfk) 36

[Atributos de ‘Petición’](#_bwliye3bwsfk) 37

[‘id’ (entero): El id único de cada petición.](#_yzsqxip0vr55) 37

[‘name’ (string): Nombre del proyecto de la petición.](#_quz1wdq78e3h) 37

[‘desarrollador\_id’ (entero): El id del desarrollador.](#_quezu2kn766u) 37

[‘cliente\_id’ (entero): El id del cliente.](#_cal3vhl19g7j) 37

[‘resumen’ (texto): El cuerpo de la petición.](#_po0ura2f5nmu) 37

[‘contestado’ (char): El carácter para saber si fue contestado o aún no.](#_6v3l3wxxvd5f) 37

[‘aceptado’ (char): El carácter para saber si el desarrollador aceptó la petición.](#_w4bxqg5cflrm) 37

[Atributos de ‘Proyecto’](#_o537n6ueu50f) 37

[‘id’ (entero): El id único de cada proyecto.](#_jlkjrdsadr7x) 37

[‘desarrollador\_id’ (entero): El id del desarrollador.](#_py0vayigiz9v) 37

[‘cliente\_id’ (entero): El id del cliente.](#_ykd4lnkbu1cr) 37

[‘name’ (string): Nombre del proyecto.](#_qlc7nefclnrw) 37

[Atributos de ‘Mensajes’](#_gf0ltpj72yh3) 37

[**Catálogo de mensajes**](#_birj4s7n8ocw) **38**

[**Anexo**](#_vj0jfdn5np9q) **42**

[Metodología Coad - Yourdon](#_s9j9hkkrdg37) 42

# Introducción

Actualmente, la Cuarta Revolución Industrial, también conocida como industria 4.0, está cambiando la forma en que los negocios operan y, por lo tanto, los entornos en los que se ven obligados a competir, obligando a las industrias que quieran mantenerse en competencia en el mercado a la digitalización de los procesos industriales y la optimización de recursos. Esto representa una evolución en el manejo y procesamiento de las operaciones, garantizando la satisfacción del cliente y la personalización de los servicios.

Esta revolución está marcada por la aparición de nuevas tecnologías como la robótica, la analítica, la inteligencia artificial, la nanotecnología y el Internet of Things (IoT), entre otros. Las organizaciones deben identificar las tecnologías que mejor satisfacen sus necesidades para invertir en ellas.

La Industria 4.0 obligará a integrar elementos y tecnologías de vanguardia dentro del mundo industrial como Internet de las Cosas (IoT), Cloud Computing, Business Intelligence, Realidad Virtual/Aumentada o Ciberseguridad, entre otras. La implementación de todo ello supone un reto muy importante para las empresas.

Por un lado, tendrán que integrar todo ello en sus procesos productivos. Por otro lado, tendrán que preparar y formar a las personas para adaptarse al cambio y así permitirles trabajar en un nuevo entorno digital y totalmente conectado. Por lo tanto, la cuarta revolución industrial viene claramente impulsada por el software y es un elemento clave para la transformación digital.

Esta es una muy buena razón para considerar la actividad de desarrollo de software como una [ingeniería](https://www.ecured.cu/Ingenier%C3%ADa_de_Software). Aún no existen suficientes empresas encargadas del desarrollo de software y de las existentes, la mayoría son extranjeras. Como México tiene potencial tanto tecnológico como económico que aún no ha explotado y se encuentra a tiempo para iniciar con la nueva revolución.

# Marco teórico

* **Industria 4.0:** es la tendencia a la automatización y el intercambio de los datos en las tecnologías de fabricación mediante tres elementos principales: el Internet de las Cosas, los Sistemas Cibernéticos Físicos y la Computación en la nube. La industria está en medio de una transformación digital acelerada exponencialmente por tecnologías en pleno crecimiento.
* **Software:** Se refiere al equipamiento lógico o soporte lógico de una [Computadora](https://www.ecured.cu/Computadora) digital, y comprende el conjunto de los componentes legales necesarios para hacer posible la realización de tareas específicas.
* **Ingeniería de software:** es la aplicación de un enfoque [sistemático](https://es.wikipedia.org/wiki/Ingenier%C3%ADa), disciplinado y cuantificable al [desarrollo](https://es.wikipedia.org/wiki/Desarrollo_de_software), operación y [mantenimiento](https://es.wikipedia.org/wiki/Mantenimiento_de_software) de [software](https://es.wikipedia.org/wiki/Software), y el estudio de estos enfoques, es decir, el estudio de las aplicaciones de la ingeniería al software.​ Integra [matemáticas](https://es.wikipedia.org/wiki/Matem%C3%A1ticas), ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería.
* **Internet de las cosas:** describe un escenario en el que diversas cosas están conectadas y se comunican. Esa innovación tecnológica tiene como objetivo conectar los ítems que usamos diariamente a internet, con el objetivo de aproximar cada vez más el mundo físico al digital.
* **Conexión vertical en forma de red:** Los Sistemas CiberFísicos están interconectados entre ellos y con trabajadores, directivos, desarrolladores, proveedores, clientes y hasta con el propio producto una vez vendido, gracias al Internet de las Cosas y al Internet de los Servicios (Cloud Computing).
* **Virtualización:** El mundo real es capturado por sensores, creando una imagen virtual de la misma, que está a su vez está conectada a Modelos de Simulación, Aplicaciones de Análisis Predictivos y Software para la ayuda de toma de decisiones. Esta información basa sus fundamentos en Big Data.
* **Modularidad:** En un mercado tan cambiante, una Industria Inteligente debe adaptarse a los cambios que se producen en el mercado de forma rápida y eficiente, mientras que hacer un estudio del mercado y un cambio de producción puede llevar mínimo una semana las industrias inteligentes están preparadas para adaptarse al cambio de forma rápida y seguir las tendencias del mercado.
* **Big Data:** se trata de la gestión y el análisis de los datos, desglosando de esta forma cantidades enormes de datos que serían imposibles de procesar con las herramientas convencionales que conocemos y que son producidos gracias al Internet de las Cosas, consiguiendo de esta forma analizar tendencias y consiguiendo un salto cualitativo dentro de la marca.
* **Cloud Computing** se encarga de obtener servicios de computación a través de la red, normalmente Internet permitiendo que nuestra empresa solo utilice los recursos necesarios en el momento preciso, mejorando los costes y optimizando los presupuestos necesarios en una era tan cambiante.
* **Fábrica inteligente** es una fábrica que permite la comunicación entre los sistemas ciber físicos, el internet de las cosas y la computación en la nube para ayudar a personas y máquinas en la ejecución de sus tareas, haciendo los procesos de fabricación y producción más sencillos, ágiles y rentables.

## Industria 4.0

La era de la información ha hecho posible que podamos acceder cada vez a más información, pudiendo procesarla y compartirla a la velocidad a la que viaja una onda electromagnética. Esto ha transformado el modo en el que consumimos y el modo en el que nos relacionamos, creando grandes redes de intercambio de datos que renuevan nuestra idea de lo individual y de lo colectivo.

La industria 4.0 o cuarta revolución industrial está destinada a revolucionar la industria de la fabricación y producción gracias al Internet de las cosas, la computación en la nube, la integración de los datos y los avances tecnológicos de los sistemas de producción y fabricación.​

Esta nueva industria inteligente busca fábricas 4.0 que gracias a los ordenadores y la automatización equipados con algoritmos de aprendizaje podrán aprender y controlar toda la robótica de una empresa y fábrica sin apenas interacción de operadores humanos, creando lo que se conoce como “Fábrica inteligente“.

## Retos de la industria 4.0

**1-. Generar redes para integrar verticalmente a todos los actores de la cadena de valor:**

La transmisión automática de datos relevantes entre sistemas de la planta de producción es esencial hoy en día. También lo es la transmisión automática de datos relevantes entre departamentos diferentes de la misma empresa. Esta integración vertical en forma de red, redundará en una mayor eficiencia en el uso de recursos (humanos, materiales, energéticos).​ Algunas claves para generar estas redes de integración vertical serán:

1. **Integración IT:** En muchos casos, las infraestructuras de tecnologías de la información están muy fragmentadas, y el resultado es una red pobre. Se necesitarán sensores, sistemas de control, redes de comunicación, aplicaciones de negocios. Las empresas deben ser capaces de integrar todos estos componentes.
2. **Analítica y gestión de datos (big data):** La industria conectada 4.0 va a generar enormes cantidades de datos. Procesar y analizar este big data generará nuevos insights, apoyará la toma de decisiones, y creará ventajas competitivas. Las compañías necesitarán contratar especialistas en las áreas del análisis y la gestión de datos (data science).
3. **Aplicaciones cloud:** El cloud hosting ofrece excelentes soluciones para almacenar grandes cantidades de datos. Las soluciones cloud serán cada vez más cruciales en las fábricas 4.0. El almacenamiento en la nube hace que cualquier información sea accesible desde cualquier lugar. Esto facilitará el acceso y el análisis de datos no sólo entre fábricas, sino a través de la cadena de valor entera.​

**2-. Integración horizontal para generar sinergias:**

Como la integración e interacción con los clientes hará que se puedan mejorar los productos y prever la demanda, se generarán grandes redes de partners de productos que son complementarios, compartiendo la tremenda información generada (también llamada “memoria del producto”), y creando valor para el usuario y para las empresas.​

Estas serán algunas de las claves de la futura integración horizontal:

1. **Optimización de modelos de negocio:** La cuarta revolución industrial generará aproximaciones radicalmente diferentes a los modelos de negocio de hoy en día, y no solo representará una mejora incremental de lo existente. Para lograrlo, las compañías necesitarán desarrollar nuevas habilidades individuales y capacidades colectivas.
2. **Cadenas de valor inteligentes:** ​Los nuevos modelos de negocio estarán más enfocados en las necesidades individuales de cada cliente, y en el crecimiento de la cooperación con otros actores. Esto hará que aparezcan nuevas necesidades en la cadena de suministro.
3. **Ciberseguridad:** ​La extensa red de conexiones y la gran cantidad de datos que implica la Industria conectada 4.0 van a aumentar dramáticamente las necesidades de cíber seguridad, ya que las actividades más importantes de las empresas cada vez van a estar más ligadas a dicho almacenamiento y transmisión de información, con lo cual la protección de dichas actividades será clave.

**3-.** **Diseño e ingeniería presente en toda la cadena de valor:**

Con todos los actores de la cadena de valor conectados en tiempo real, el rediseño, la personalización y la optimización de la ingeniería serán un flujo continuo y retroalimentado, ya que la información de toda la cadena de valor estará llegando constantemente. En lugar de ser recabada periódicamente mediante técnicas poco fiables como los cuestionarios. Algunas de las claves serán:

1. **Entender los 10 tipos de innovación:** La nueva era industrial 4.0 permitirá que la innovación salga de sus áreas tradicionales (el producto), para polinizar otras áreas de la empresa.
2. **Gestionar la innovación de forma eficiente:** ​La gestión de la innovación implica a la compañía entera, desde la estrategia, la organización, las líneas de negocios y el desarrollo de productos. Las tecnologías de la información afectarán profundamente al desarrollo de productos, haciéndolo mucho más ágil y enfocado a las necesidades de cada cliente individual.
3. **Comprender y gestionar mejor el ciclo de vida​:** ​El Big Data y el Análisis Predictivo generarán indicadores capaces de anticipar la evolución de los productos. Aplicaciones de Inteligencia Artificial, en disposición de toda la información, generarán las bases para la toma de decisiones.

**4-. Aceleradores tecnológicos**

Ya hemos mencionado ampliamente aquellas tecnologías existentes que darán forma al mundo y a la industria que está por venir. La buena comprensión y uso de dichos aceleradores tecnológicos permitirán soluciones individualizadas, más flexibilidad, ahorro de costes, y una total interconexión de toda la cadena de valor. Algunas de las claves para aprovechar estos aceleradores tecnológicos serán las siguientes:​

1. **Corporate Venturing​:** Las empresas emplearán una mayor parte de sus recursos a la inversión en startups, externalizando de algún modo la innovación, a través de la inversión en empresas que investigan tecnologías disruptivas. Esto estaría en consonancia con las tesis de Clayton M. Christensen en su obra de referencia “El dilema del innovador”.​
2. **Aprendizaje organizacional​:** La Big Data y el Análisis Predictivo asegurará que la organización aprenda y no repita los sesgos propios de los seres humanos en la toma de decisiones, obligando a los directivos a observar la realidad con mucha más objetividad.​

Una vez que se entendió el concepto de Industria 4.0 y la importancia que tendrá en los siguientes años para el desarrollo global del mercado y los países, hablaremos acerca del desarrollo de software y el papel que jugará a lo largo de esta Revolución.

Cuando se va desarrollar un software intervienen muchas personas como lo es el cliente que es el que tiene el problema y desea que sea solucionado, para esto existe el [Analista de Sistema](https://www.ecured.cu/index.php?title=Analista_de_Sistema&action=edit&redlink=1) que es el encargado de hacerle llegar todos los requerimientos y necesidades que tiene el cliente a los programadores que son las personas encargadas de realizar lo que es la codificación y [diseño](https://www.ecured.cu/Dise%C3%B1o) del sistema para después probarlo y lo instalan al cliente.

El desarrollo de software comprende de las siguientes etapas que al finalizarlas:

1. **Análisis de requisitos:** Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios.
2. **Diseño y arquitectura:** Se refiere a determinar cómo funcionará de forma general sin entrar en detalles. Se definen los casos de uso para cubrir las funciones que realizará el sistema, y se transforman las entidades definidas en el análisis de requisitos en clases de diseño, obteniendo un modelo cercano a la programación orientada a objetos.
3. **Programación:** Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no es necesariamente la porción más larga. La complejidad y la duración de esta etapa está íntimamente ligada al o a los lenguajes de programación utilizados.
4. **Pruebas:** Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó.
5. **Mantenimiento:** Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Una pequeña parte de este trabajo consiste en arreglar errores, o bugs. Se puede decir que con la mejora continua garantiza la calidad del producto, ya que el estar aplicando día con día es la mejor decisión que puede llegar a tener cualquier empresa, porque de esta manera evita grandes problemas en la elaboración o desarrollo de los productos.

# Estado del arte

¿Qué es SOFTECH? Es un Sistema de Apoyo Web para solicitar / implementar proyectos de Ingeniería y desarrollo de Software. Originalmente, está planeado únicamente para clientes con necesidades de software y desarrolladores de la misma área. Dependiendo de la aceptación del público, la usabilidad y los ingresos económicos, podríamos ampliar nuestros servicios.

* Solicita proyectos referentes al desarrollo de software.
* Desarrolla proyectos de tu área de desempeño.
* Controla y administra el ciclo de vida del desarrollo de software.

Actualmente y como se mencionó anteriormente, no existen suficiente empresas encargadas al desarrollo de software en México que pueda satisfacer la demanda de la nueva industria. A continuación, se mencionara las 5 empresas más importantes:

1. **Danthop:** es una empresa joven que apenas se esté posicionando en internet, ha logrado ganar muy buen mercado y tener un alto impacto en todo el país. Se posiciona en el número uno no por sus utilidades, sino por su alta eficiencia, bajos costos y excelente servicio al cliente. Su éxito se debe a la visión de su fundador ante los nuevos mercados móviles emergentes.
2. **Vexilo:** Esta empresa se formó a finales del 2009 con un equipo de profesionales, actualmente, se ha convertido en una de las empresas más reconocidas en el medio; logrando romper paradigmas sobre las compañías de desarrollo de software en México y empezando su expansión a otros países. Una mejor forma de hacer las cosas en el desarrollo de software.
3. **Bittech:** La oportunidad de su futuro, la ven en el éxito de sus clientes. Ellos son una empresa 100% mexicana de consultoría en tecnologías de información, especializada en el desarrollo de aplicaciones y soluciones de negocio para medianas y grandes empresas en los sectores educativo, industrias diversas, empresas de servicio y gobierno.
4. **HighBits:** Satisface las necesidades en tecnologías de la información de sus clientes. HighBits es una compañía mexicana especializada en el ramo de los servicios tecnológicos para el sector público, académico y privado.
5. **Nationalsoft:** Su compromiso de calidad los ha llevado a obtener certificaciones. Ellos son una empresa líder dedicada al desarrollo y comercialización de software, ubicada en la ciudad de Mérida, Yucatán, México. Actualmente cuentan con presencia comercial en México y varios países de habla hispana a través de sus representantes.

# Métricas de estimación

# 

## Puntos de función

|  |  |  |  |
| --- | --- | --- | --- |
| Componente | Tipo de componente | Nivel de complejidad | P.F |
| Registro de nuevo cliente/desarrollador | Entrada externa | Bajo | 4 |
| Listado de desarrolladores | Consulta externa | Medio | 5 |
| Tabla de usuarios | Archivo lógico interno | Medio | 7 |
| Subida de evidencias (desarrollador) | Entrada externa | Alto | 7 |
| Visualización de avances del desarrollador | Salida externa | Medio | 10 |
| Conversación cliente-desarrollador | Entrada externa  Salida externa | Alto | 22 (15+7) |
| **Total** |  |  | 55 |

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables? (0: NO)

2. ¿Se requiere comunicación de datos? (5: Esencial)

3. ¿Existen funciones de procesamiento distribuido? (0: NO)

4. ¿Es crítico el rendimiento? (4: Significativo)

5. ¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado? (5: Significativo)

6. ¿Requiere el sistema entrada de datos interactiva? (4: Esencial)

7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones? (5: Si)

8. ¿Se actualizan los archivos maestros de forma interactiva? (0: No)

9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones? (3: Moderadamente)

10. ¿Es complejo el procesamiento interno? (3: Moderadamente)

11. ¿Se ha diseñado el código para ser reutilizable? (0: No)

12. ¿Están incluidas en el diseño la conversión y la instalación? (0: No)

13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones? (5: Si)

14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario? (3: Moderadamente)

Suma total: 37

PF: 55 \* (0.65 + 0.001\*37) = **37.77,** con una productividad de 9.9.

Puntos de función: 55 ± 20 % (20 % pues se estima que puedan haber cambios al implementar funciones a futuro) nos da como resultado 44 y 66, con lo cual podemos deducir que la complejidad del proyecto es mediana.

# Cronograma

|  |  |  |  |
| --- | --- | --- | --- |
| **Actividad** | **Fecha de inicio** | **Duración** | **Fecha de término** |
| **Entrega de propuesta** | 01/04/2019 | 1 | 01/04/2019 |
|  |  |  |  |
| **1-. Análisis** | 02/04/2019 | 15 | 16/04/2019 |
| Identificación de requerimientos | 02/04/2019 | 4 | 05/04/2019 |
| Identificación de casos de uso | 06/04/2019 | 3 | 08/04/2019 |
| Modelado de procesos | 09/04/2019 | 4 | 12/04/2019 |
| Escritura de casos de uso | 13/04/2019 | 4 | 16/04/2019 |
|  |  |  |  |
| **2-. Diseño** | 17/04/2019 | 5 | 21/04/2019 |
| Base de datos | 17/04/2019 | 5 | 21/04/2019 |
| Aplicación Web | 17/04/2019 | 5 | 21/04/2019 |
| Definición de arquitectura | 17/04/2019 | 5 | 21/04/2019 |
|  |  |  |  |
| **3-. Implementación** | 22/04/2019 | 25 | 17/05/2019 |
|  |  |  |  |
| **4-. Pruebas** | 18/05/2019 | 5 | 22/05/2019 |
|  |  |  |  |
| **5-. Retroalimentación** | 22/05/2019 | 6 | 28/05/2019 |
|  |  |  |  |
| **6-. Despliegue** | 28/05/2019 | 4 | 31/05/2019 |

**Tabla 1. Cronograma de actividades.**

# Metodología

## 

## Metodología SCRUM

La metodología Scrum es tendencia en la gestión de proyectos. El sector del desarrollo de software es el principal representante de este tipo de metodología ágil. Se trata de planificar tus proyectos en pequeños bloques o Sprints, e ir revisando y mejorando el anterior.

Definición. Scrum es un método para trabajar en equipo a partir de iteraciones o Sprints. Así pues, **Scrum es una metodología ágil**, por lo que su objetivo será controlar y planificar proyectos con un gran volumen de cambios de última hora, en donde la incertidumbre sea elevada.

Se suele planificar por semanas. Al final de cada **Sprint o iteración**, se va revisando el trabajo validado de la anterior semana. En función de esto, se priorizan y planifican las actividades en las que invertiremos nuestros recursos en el siguiente Sprint.

La metodología Scrum se centra en ajustar sus resultados y responder a las exigencias reales y exactas del cliente. De ahí, que se vaya revisando cada entregable, ya que los requerimientos van variando a corto plazo. El tiempo mínimo para un Sprint es de una semana y el máximo es de cuatro semanas.

## 

## Roles de Scrum

La metodología Scrum tiene unos roles y responsabilidades principales, asignados a sus procesos de desarrollo. Estos son:

* **Project Owner**. Se asegura de que el proyecto se esté desarrollando acorde con la estrategia del negocio. Escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.
* **Master Scrum**. Elimina los obstáculos que impiden que el equipo cumpla con su objetivo.
* **Development team Member**. Los encargados de crear el producto para que pueda estar listo con los requerimientos necesarios. Se recomienda que sea un equipo multidisciplinar, de no más de 10 personas.

## 

## Fases de la metodología Scrum

El desarrollo de producto tiene un ciclo de vida en la metodología Scrum. Estas son fases en las que se divide un proceso Scrum:

* ¿Qué y quién? El producto que queremos conseguir y los roles de equipo.
* ¿Dónde y cuándo? El plazo y el contenido del Sprint.
* ¿Por qué y cómo? Las distintas herramientas para aplicar esta metodología ágil.

Cada Sprint puede tener una serie de eventos o etapas. Los más comunes son:

1. Reunión para la planificación del Sprint. En ella, se divide el tiempo de duración del Sprint, así como el objetivo y entregable del mismo. Además, el equipo de desarrollo deberá saber cómo realizarlo.
2. Trabajo de desarrollo durante el Sprint. Nos aseguramos que los objetivos se están cumpliendo, que no se producen cambios que alteran el objetivo del Sprint y se mantiene un feedback constante con el cliente o dueño del proyecto.
3. Revisión del Sprint. Reunión con el cliente o dueño del proyecto, en la que se estudia y revisa el Product Backlog del Sprint. Se definen los aspectos a cambiar, en caso necesario, de mayor valor o probables para planificarlo en el siguiente Sprint.
4. Retrospectiva del proyecto. Oportunidad del equipo de desarrollo para mejorar su proceso de trabajo y aplicar los cambios en los siguientes Sprints.

## Beneficios

* **Cumplimento de expectativas:** El cliente establece sus expectativas indicando el valor que le aporta cada requisito / **historia** del proyecto, el equipo los estima y con esta información el **Product Owner** establece su prioridad.
* **Flexibilidad a cambios:** Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos.
* **Mayor calidad del software:** La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
* **Mayor productividad:** Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.

# Modelo de procesos

## Basado en prototipos

También conocido como desarrollo con prototipación o modelo de desarrollo evolutivo, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición.

Este modelo se utilizan para dar al usuario una vista preliminar de parte del software. Este modelo es básicamente prueba y error ya que si al usuario no le gusta una parte del prototipo significa que la prueba fallo por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho.

Además el prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar mucho dinero pues a partir de que este sea aprobado nosotros podemos iniciar el verdadero desarrollo del software. Pero eso si al construir el prototipo nos asegura que nuestro software sea de mejor calidad, además de que su interfaz sea de agrado para el usuario. Un prototipo podrá ser construido sólo si con el software es posible experimentar.

## Etapas

* Recolección y refinamiento de requisitos
* Modelado, diseño rápido
* Construcción del Prototipo
* Desarrollo, evaluación del prototipo por el cliente
* Refinamiento del prototipo
* Producto de Ingeniería

## Ventajas

* No modifica el flujo del ciclo de vida
* Reduce el riesgo de construir productos que no satisfagan las necesidades
* Reduce costo y aumenta la probabilidad de éxito
* Exige disponer de las herramientas adecuadas
* Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
* También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

# Modelo de negocios

## 

## Long Tail

Se trata de un concepto acuñado por Chris Anderson en su artículo The Long Tail, que luego se convirtió en el libro La economía Long Tail. Sus orígenes se basan en una idea simple, y es que tradicionalmente hemos diseñado todos los modelos de negocio alrededor de la regla de Pareto: **el 20% de los productos generarán el 80% de las ventas.** O dicho de otro modo, centramos nuestros esfuerzos en los mercados de masas, en **vender todavía más los productos que tienen mucha demanda**.

Si representamos en una gráfica la **distribución de ventas de cualquier empresa** tendrá esta pinta seguramente:

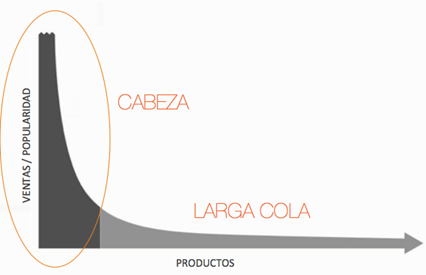


Imagen 1. Modelo de negocios Long Tail

* **CABEZA**: Estos son los productos más populares, los que más habitualmente piden los clientes. Como dicen los ingleses, el mainstream.
* **COLA**: Aquí se acumula una serie muy diversa de productos que aunque SI tiene demanda, es mucho menor que en el caso de la cabeza y habitualmente resultan descartados.

**¿Y qué nos dice la estrategia empresarial?** Pues lo que parece obvio, que centramos nuestros esfuerzos en la cabeza de la cola, que es donde es más sensato invertir nuestros recursos para **vender más**.

Sin embargo en los últimos tiempos han surgido nuevos modelos, denominados “Long Tail” o de “Larga cola” que basan su efectividad en **vender menos unidades de más cosas** (o lo que es lo mismo, centrar el foco en vender un número mayor de productos (el 80%) con menor volumen de ventas). Y esto sucede porque cada vez más **los consumidores prefieren productos o servicios de nicho, más específicos y que satisfacen mejor sus necesidades**, qué productos generalistas.

¿Que tener en cuenta en un modelo de negocio de larga cola?

Para crear un modelo long tail es muy importante **comprender cómo funciona**, así como los principales **condicionantes** del mismo. Y sobre todo, sus tres principales palancas:

1. **Demanda:** La mejor forma de que sea realmente rentable vender menos unidades de muchas cosas es que seamos capaces de agregar mucha demanda. Es decir, el éxito de estos modelos depende, principalmente de que podamos vender muchas veces cosas que por separado tienen poca demanda.
2. **Almacenamiento:** La decisión de centrar nuestra atención en la cabeza de la cola parte del hecho exclusivo de que tenemos un recurso muy caro y escaso (el espacio de almacenamiento), y que queremos maximizar su rentabilidad.  
   Sin embargo, si encontramos una forma de disminuir de forma agresiva su coste y distribución (a través de estrategias de optimización de espacio, reducción de costes por ubicación, logísticas muy ajustadas, producción bajo demanda…etc) seremos capaces de aprovechar más espacio para vender cosas que tienen menor demanda.
3. **Recomendación:** La tercera palanca sobre la que debemos actuar en un modelo de negocio de *larga cola* es a través del uso de estrategias de recomendación.  
   Otra opción ligeramente diferente es utilizar lo que yo llamo ganchos, algo que tiene sentido en modelos muy enfocados a long tail: aunque de donde saquemos rentabilidad es de la cola, es difícil sostener un modelo solo con esa demanda.

# Técnicas de recolección de datos

La recolección de datos se refiere al uso de una gran diversidad de técnicas y herramientas que pueden ser utilizadas por el analista para desarrollar los sistemas de información, los cuales pueden ser la entrevistas, la encuesta, el cuestionario, la observación, el diagrama de flujo y el diccionario de datos.

Todos estos instrumentos se aplicarán en un momento en particular, con la finalidad de buscar información que será útil a una investigación en común. En la presente investigación trata con detalle los pasos que se debe seguir en el proceso de recolección de datos, con las técnicas ya antes nombradas.

Existen 5 técnicas de recolección de datos, pero nuestro equipo utilizó sólo 2:

* **ENTREVISTA:** La entrevista es una conversación dirigida, con un propósito específico y que usa un formato de preguntas y respuestas.Se establece así un diálogo, pero un diálogo peculiar, asimétrico, donde una de las partes busca recoger informaciones y la otra se nos presenta como fuente de estas informaciones.

Una entrevista es un diálogo en el que la persona (entrevistador), generalmente un periodista hace una serie de preguntas a otra persona (entrevistado), con el fin de conocer mejor sus ideas, sus sentimientos su forma de actuar.

**Preparación de la Entrevista**

1. Determinar la posición que ocupa de la organización el futuro entrevistado, sus responsabilidades básicas, actividades, etc. (Investigación).
2. Preparar las preguntas que van a plantearse, y los documentos necesarios (Organización).
3. Fijar un límite de tiempo y preparar la agenda para la entrevista. (Sicología).
4. Elegir un lugar donde se puede conducir la entrevista con la mayor comodidad (Sicología).
5. Hacer la cita con la debida anticipación (Planeación).

**Tipo de preguntas**

* **Preguntas abiertas:** Son aquellas preguntas que describen hechos o situaciones por parte del entrevistado con una gran cantidad de detalles que a juicio del entrevistado son importantes.
* **Preguntas cerradas:** En las preguntas cerradas las respuestas posibles están cerradas al entrevistado, debido a que solamente puede responder con un número finito, tal como “ninguno”, “uno”, o “quince”. Una pregunta cerrada limita las respuestas disponibles al entrevistado.
* **CUESTIONARIO:** Los cuestionarios proporcionan una alternativa muy útil para la entrevista; sin embargo, existen ciertas características que pueden ser apropiada en algunas situaciones e inapropiadas en otra. Al igual que la entrevistas, deben diseñarse cuidadosamente para una máxima efectividad. Selección de formas para cuestionarios.

El desarrollo y distribución de los cuestionarios; por lo tanto, el tiempo invertido en esto debe utilizarse en una forma inteligente. También es importante el formato y contenido de las preguntas en la recopilación de hechos significativos.

Existen dos formas de cuestionarios para recabar datos: cuestionarios abiertos y cerrados, y se aplican dependiendo de si los analistas conocen de antemano todas las posibles respuestas de las preguntas y pueden incluirlas. Con frecuencia se utilizan ambas formas en los estudios de sistemas.

* **Cuestionario Abierto:** los cuestionarios pueden ser abiertos y se aplican cuando se quieren conocer los sentimientos, opiniones y experiencias generales; también son útiles al explorar el problema básico, por ejemplo, un analista que utiliza cuestionarios para estudiar los métodos de verificación de crédito, es un medio.

El formato abierto proporciona una amplia oportunidad para quienes respondan escriba las razones de sus ideas. Algunas personas sin embargo, encuentran más fácil escoger una de un conjunto de respuestas preparadas que pensar por sí mismas.

* **Cuestionario Cerrado:** El cuestionario cerrado limita las respuestas posibles del interrogado. Por medio de un cuidadoso estilo en la pregunta, el analista puede controlar el marco de referencia. Este formato es el método para obtener información sobre los hechos. También fuerza a los individuos para que tomen una posición y forma su opinión sobre los aspectos importantes.

# Requerimientos

# 

|  |  |
| --- | --- |
| **Clave** | **Descripción** |
| **RN01** | Cualquier usuario puede registrarse como de cliente o desarrollador. |
| **RN02** | Todo usuario registrado como cliente, tiene autorizado contratar uno o más servicios mientras cuente con el crédito suficiente. |
| **RN03** | Todo usuario registrado como desarrollador tiene autorizado trabajar en uno o más servicios mientras su desempeño sea eficiente. |
| **RN04** | Todo usuario registrado como cliente, tiene derecho a cancelar su proyecto sin hacerse acreedor a una penalización en un lapso no mayor a 5 días a partir de la fecha de inicio estipulada en el contrato. |
| **RN05** | Todo usuario registrado como desarrollador, tiene derecho a cancelar su proyecto sin hacerse acreedor a una penalización en un lapso no mayor a 3 días a partir de la fecha de inicio estipulada en el contrato. |
| **RN06** | Cualquier usuario (Cliente o desarrollador) tiene derecho a cancelar un proyecto, fuera del tiempo estipulado en las RN04 y RN05, sin hacerse acreedor a una penalización, exponiendo motivos y bajo previa supervisión de la administración |
| **RN07** | Todo usuario registrado como Cliente, será penalizado en caso de cancelar el proyecto fuera del tiempo estipulado en la RN04. La penalización consistirá en el pago total del proyecto y una disminución de puntaje en nuestra aplicación |
| **RN08** | Todo usuario registrado como Desarrollador, será penalizado en caso de cancelar el proyecto fuera del tiempo estipulado en la RN05. La penalización consistirá en una multa igual al pago total del proyecto y una disminución de puntaje en nuestra aplicación. |
| **RN09** | Todo usuario registrado como moderador, tiene autorizado verificar el cumplimiento del contrato y establecer las multas pertinentes, en caso de que un punto no sea cubierto. |

# Requerimientos Funcionales

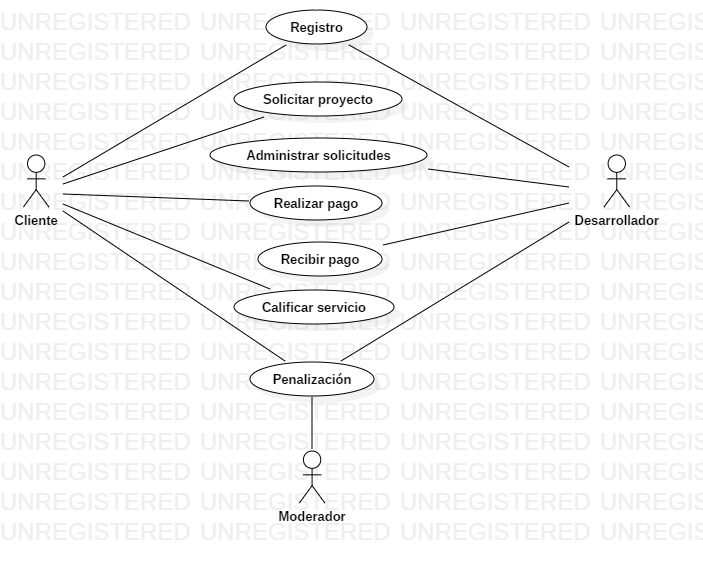
|  |  |  |
| --- | --- | --- |
| **Clave** | **Nombre** | **Descripción** |
| **RF01** | Registro de usuarios | Se permitirá el registro de nuevos usuarios o el acceso de ya existentes. |
| **RF02** | Administrar usuarios | El sistema permitirá registrar clientes o desarrolladores, dependiendo del papel que el usuario quiera desempeñar; así como el ingreso de sus datos personales para crear un perfil. |
| **RF03** | Solicitar proyecto | El cliente podrá solicitar un proyecto y describir de manera detallada la problemática a resolver. |
| **RF04** | Administrar peticiones | El desarrollador recibirá las solicitudes de los clientes, para así proceder a aceptar y ponerse en contacto con él cliente. |
| **RF05** | Contacto entre usuarios | Los usuarios recibirán los datos tanto del cliente como del desarrollador, respectivamente. |

# 

# Requerimientos no Funcionales

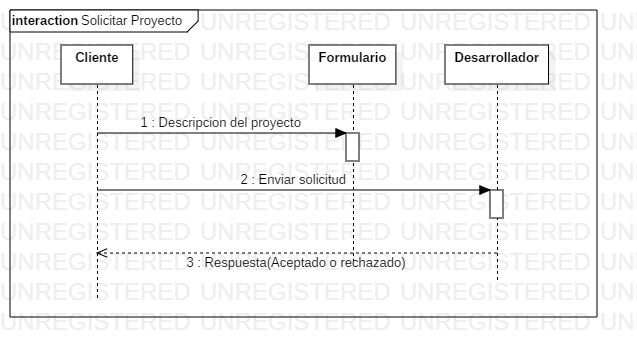
|  |  |  |
| --- | --- | --- |
| **Clave** | **Nombre** | **Descripción** |
| **RNF01** | Funcionalidad general | El sistema debe ser fácil de utilizar tanto para desarrolladores como para clientes. |
| **RNF02** | Concurrencia | El sistema debe soportar la cantidad de visitantes sin importar el tipo que sean. |
| **RNF03** | Rapidez | La solicitud del cliente debe atenderse lo más pronto posible y no demorarse para no perderlo. |
| **RNF04** | Manejo de errores | En caso de existir errores como de consulta o búsqueda de información, desplegar mensajes en vez de terminar el programa. |
| **RNF05** | Disponibilidad | El sistema se debe poder visualizar en los navegadores más comunes (Chrome, Firefox y Opera) |

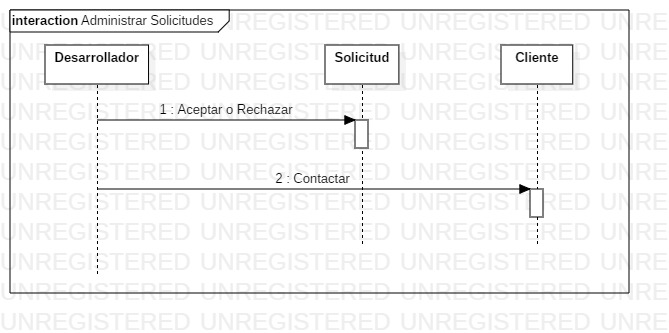
## Diagrama de casos de uso

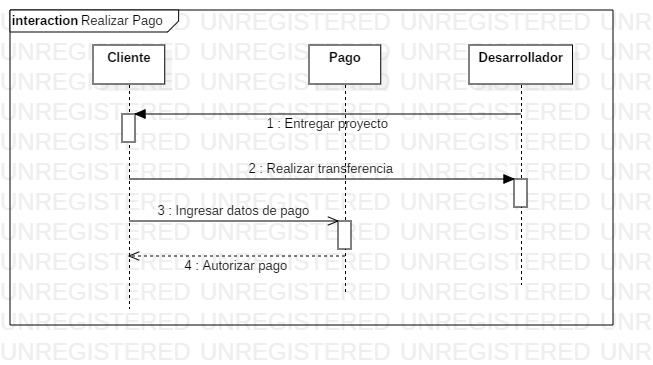


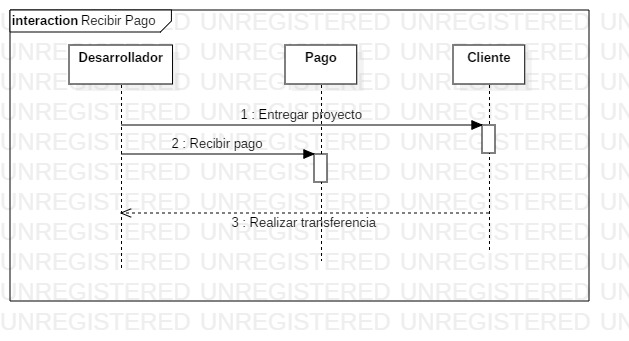
## Diagrama de secuencias

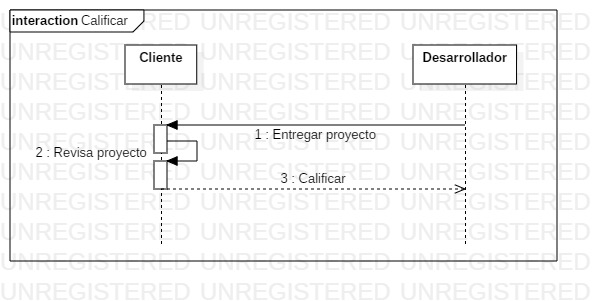
## 

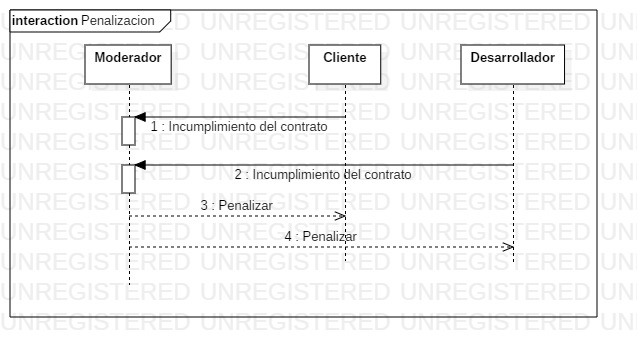












## Diagrama de clases

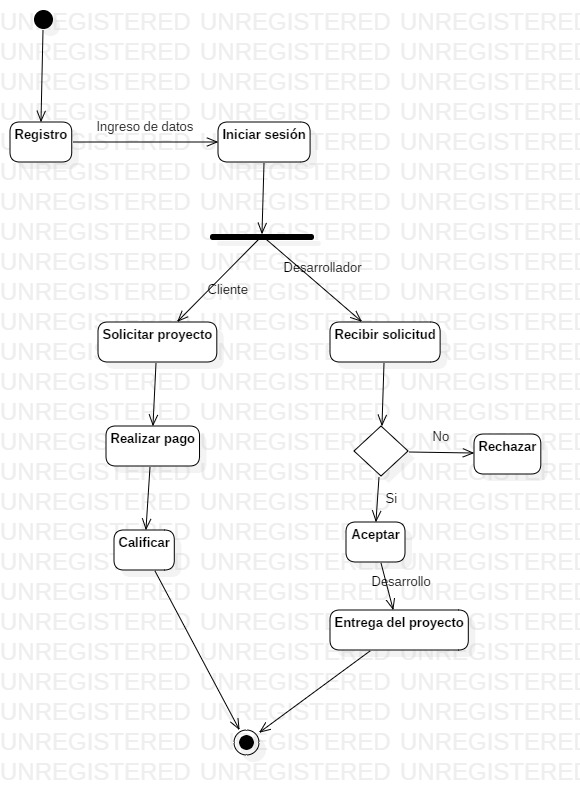
## 

## 

## 

## 

## Diagrama de actividades

****

# 

# Pruebas

## Prueba de integración

* **Objetivo:** Verificar que los componentes del software trabajen eficientemente en conjunto.
* **Descripción:** La prueba de integración es una extensión lógica de las pruebas unitarias. Dos unidades que ya han sido testeadas y combinadas en un componente y su interfaz son testeadas entre ellas. Un componente, en este ejemplo, se refiere a un agregado que está integrado en más de una unidad, estas son combinadas en componentes, que son agregadas por orden en partes más grandes del programa. El motivo de las pruebas de integración es de verificar la funcionalidad y la seguridad entre los componentes que han sido integrados. Identifica los problemas que ocurren cuando las unidades se combinan. Esto es particularmente beneficioso porque determina cómo de eficientes son las pruebas trabajando juntos. Es importante conocer que las pruebas de integración están basadas en pruebas unitarias con base de datos u otra biblioteca ajena de terceras partes.
* **Softwares por utilizar:** Mocha, Jasmine, Jest y Ava
* **Resultados:** Esperamos ver que cada uno de nuestros componentes funcionen bien en conjunto.

## 

## Mantenibilidad

* **Objetivo:** Saber si nuestro software es mantenible.
* **Descripción:** Para saber si un software es mantenible se debe ser de fácil entendimiento a parte de tener la caracterı́stica de que sea fácil de modificar usando técnicas de programación para este objetivo. También debe tener una complejidad ciclomática baja para que ası́ no haya tantos errores el código
* **Softwares por utilizar:** Sonar
* **Resultados:** Muy muy pocos errores en sonar sin ser crı́tico alguno y una complejidad ciclomática por debajo de 15.

## Pruebas de estrés

* **Objetivo:** Saber cuantos usuarios soporta nuestro servidor.
* **Descripción:** Esta evaluación pone a prueba la robustez y la confiabilidad del software sometiéndose a condiciones de uso extremas. Entre estas condiciones se incluyen el envı́o excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan bugs (defectos) potencialmente peligrosos.
* **Softwares por utilizar:** Jmeter, Gatling
* **Resultados:** Que nuestro servidor sirva para al menos 50 personas en el sistema al mismo tiempo sin que existan errores.

## Pruebas de portabilidad

* **Objetivo:** Probar que nuestro software sea de fácil instalación y pueda servir en todos los dispositivos para el cual fue diseñado.
* **Descripción:** Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta caracterı́stica se subdivide a su vez en las siguientes subcaracterı́sticas:
  + Adaptabilidad. Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
  + Capacidad para ser instalado. Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
  + Capacidad para ser reemplazado. Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.
* **Resultados:** Que el sistema sirva en los dispositivos móviles con Android desde la primera versión que elegimos hasta la más actual.

# Tecnologías a utilizar

# 

La arquitectura para la parte web debe ser MVC

* **HTML5:** Es un lenguaje sencillo, fácil de aprender y fácil de leer e interpretar. Su sencillez hace que pueda diseñarse, implementarse y diseñar un sitio web en muy poco tiempo o en una menor cantidad de tiempo a diferencia de otros sitios web. Es el lenguaje más extendido, todos los navegadores lo admiten.
* **CSS:** Con una Hoja de Estilo podemos modificar la presentación de cada elemento sin modificar el código HTML. Así, el mantenimiento del sitio web se hace más sencillo. Las Hojas de Estilo pueden usarse con otros lenguajes de programación (por ejemplo, JavaScript) para conseguir efectos dinámicos en las páginas.
* **Bootstrap:** Puedes tener una web bien organizada de forma visual rápidamente: la curva de aprendizaje hace que su manejo sea asequible y rápido si ya sabes maquetar. Permite utilizar muchos elementos web: desde iconos a desplegables, combinando HTML5, CSS y Javascript. El diseño será adaptable, no importa el dispositivo, la escala o resolución.
* **Laravel:** es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP5 y PHP7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP. Gran parte de Laravel está formado por dependencias, especialmente de Symfony, esto implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.

# Modelo de entidades del negocio

## Descripción de entidades del negocio

### Atributos de ‘Usuario’

* ‘id’ (entero): Un número que inicia desde 1 y es diferente en cada usuario, llave principal para encontrar un usuario.
* ‘tipo’ (string): El tipo de usuario que pertenece a nuestra empresa (cliente, desarrollador y moderador).
* ‘nombre’ (string): Nombre o apodo como vamos a llamar al usuario.
* ‘email’ (string): Es el email donde van a llegar los correos y notificaciones del usuario.
* ‘foto’ (imagen): En el caso de los desarrolladores si les damos la opción de cargar una imagen para su perfil y la información que ven los clientes.
* ‘descripcion’ (string): Un pequeño resumen de cómo son como desarrolladores y porque elegirlos.
* ‘calificación’ (entero positivo): La calificación que se le da como desarrolladores y como usuarios.
* ‘password’ (string encriptado): La contraseña del usuario.
* ‘especialidad’ (string): Una de nuestras muchas especialidades de desarrollod de software.

### Atributos de ‘Petición’

### ‘id’ (entero): El id único de cada petición.

### ‘name’ (string): Nombre del proyecto de la petición.

### ‘desarrollador\_id’ (entero): El id del desarrollador.

### ‘cliente\_id’ (entero): El id del cliente.

### ‘resumen’ (texto): El cuerpo de la petición.

### ‘contestado’ (char): El carácter para saber si fue contestado o aún no.

### ‘aceptado’ (char): El carácter para saber si el desarrollador aceptó la petición.

### Atributos de ‘Proyecto’

### ‘id’ (entero): El id único de cada proyecto.

### ‘desarrollador\_id’ (entero): El id del desarrollador.

### ‘cliente\_id’ (entero): El id del cliente.

### ‘name’ (string): Nombre del proyecto.

* ‘avance\_1’ (fecha): La fecha del primer avance.
* ‘entrega\_1’ (char): El carácter que dice si entregó el desarrollador la primera entrega o no.
* ‘avance\_2’ (fecha): La fecha del segundo avance.
* ‘entrega\_2’ (char): El caracter que dice si entregó el desarrollador la segunda entrega o no.
* ‘avance\_final’ (fecha): La fecha del proyecto terminado.
* ‘entrega\_final’ (char): El caracter que dice si entregó el desarrollador hizo una entrega final o no.

### Atributos de ‘Mensajes’

* ‘id’ (entero): El id único de cada mensaje.
* ‘remitente’ (entero): El id de quien mandó el mensaje.
* ‘destinatario’ (entero): El id de quien recibe el mensaje.
* ‘mensaje’ (texto): El cuerpo del mensaje.
* ‘project\_id’ (entero sin signo): El id del proyecto al que pertenece el mensaje.

# 

# Catálogo de mensajes



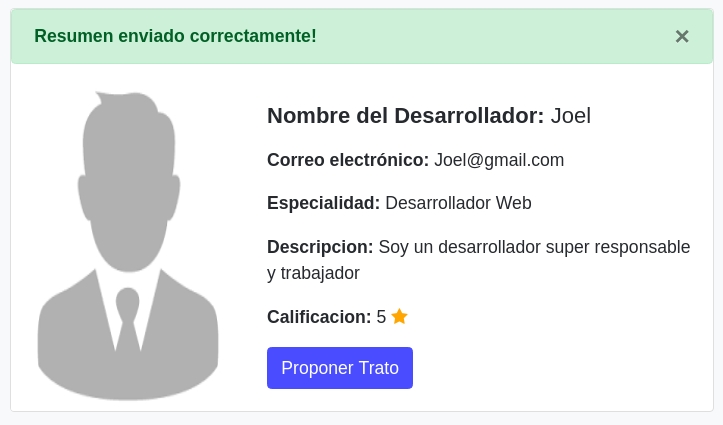
Descripción: El mensaje de validación para llenar los datos de inicio de sesión.

## 

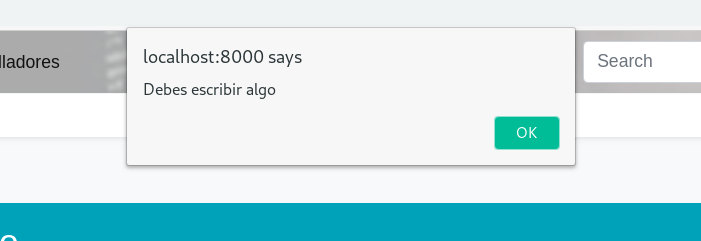
Descripción: Los mensajes de validación al momento de registrar un desarrollador son estos.

## 

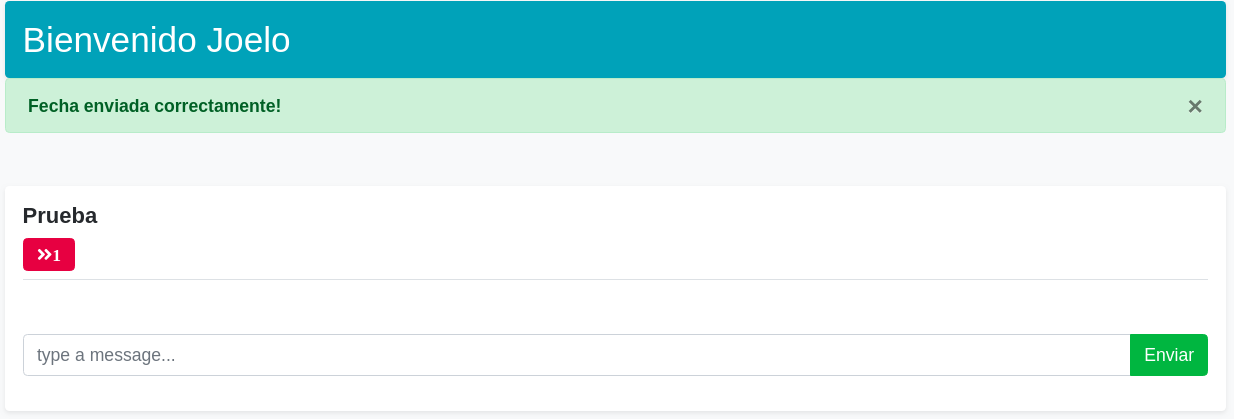
Descripción: Los mensajes de validación al momento de registrar un cliente son estos.



Descripción: El mensaje que regresa al momento de enviar una propuesta correctamente.



Descripción: Mensaje de error cuando no escribes nada en un mensaje.

Descripción: Mensaje de éxito cuando se propone una fecha para un avance.

# Anexo

## Metodología Coad - Yourdon

HO Objetos HO Operaciones

La primera acción que necesita realizar el **usuario**, es **registrarse** en nuestro sistema, esto con la finalidad de poder llevar un mejor control de usuarios y que nuestro sistema no pierda seriedad. Existirán 4 tipos de usuarios: **cliente**, **desarrollador**, **administrador** y **moderador**.

El cliente tendrá la oportunidad de **solicitar el proyecto** que él necesite dependiendo de sus necesidades. El lanzará la petición y el sistema **listará** a los desarrolladores disponibles que puedan realizar su petición.

Una vez que el cliente haya seleccionado al desarrollador, se **enlazaran** para que ambos usuarios estén relacionados en el proyecto y se **establecerá** el **contrato**.

Paralelamente a estas operaciones, también se **asignará** un moderador a cada proyecto, esto con la finalidad de verificar que ambas partes cumplan con su parte del contrato.

La operación principal, es el **pago de servicio** que se realizará durante el desarrollo del proyecto; no sólo por el desarrollo, también puede ser algún tipo de penalización.

Además, los usuarios (Clientes y Desarrolladores) podrán **acceder** a la información de los demás usuarios, para conocer su historial dentro de nuestro sistema y poder valorar la situación.

Una vez terminado el contrato, sin importar el caso; ambos usuarios (Clientes y Desarrolladores), podrán **calificar** a su contraparte; esta calificación primero será válida por el moderador para evitar difamación por alguna de las partes involucradas.

|  |  |  |
| --- | --- | --- |
| **Objeto Potencial** | **Descripción** | **Status** |
| **Cliente** | Este objeto es el que solicitará los proyectos, es uno de los más importantes ya que estará involucrado en la mayoría de los procesos y tendrá acceso tanto a su información como a la del objeto desarrollador. | ACEPTADO |
| **Desarrollador** | Este objeto es el encargado de desarrollar las peticiones realizadas por el cliente, tendrá acceso tanto a su información como a la cliente con el que se encuentra trabajando. | ACEPTADO |
| **Administrador** | Este objeto es el de mayor jerarquía y el más importante ya que tendrá que supervisar todos los procesos del sistema y además tendrá acceso ilimitado a la información que se maneja dentro del mismo. | ACEPTADO |
| **Moderador** | Este objeto es una pieza fundamental dentro de las peticiones entre cliente y desarrollador. Él realizará el contrato y llevará el seguimiento del proyecto para que ambas partes cumplan con lo establecido. | ACEPTADO |
| **Contrato** | Objeto de carácter legal cuya importancia radica en establecer un lineamiento a seguir con derechos y obligaciones para ambas partes involucradas. | ACEPTADO |