



**UNIVERSIDAD POLITÉCNICA DE SINALOA
PROGRAMA ACADÉMICO DE
INGENIERÍA EN INFORMÁTICA**

Tesina

**“Desarrollo de Pagina web para la
denuncia ciudadana en faltas al reglamento
de tránsito y la educación vial”**

Para cumplir la acreditación de las estadías
profesionales y contar con los créditos necesarios para
obtener el grado de Ingeniero en Informática

Autor:

Romero Carrillo Mario Rafael


Asesor:

M.C Francisco Javier Real Angulo

Asesor OR:

ING. Domingo Marco Antonio Hernández Jiménez

Mazatlán, Sinaloa 15 de diciembre de 2018



Zapopan, Jalisco, 12 de Diciembre del 2018

LIC. EDUARDO CRESPO CAMPA.
DIRECTOR DE VINCULACIÓN, DIFUSIÓN Y EXT. UNIVERSITARIA.
UNIVERSIDAD POLITÉCNICA DE SINALOA.

PRESENTE

Por este medio, hago de su conocimiento que el alumno(a) el C. **Mario Rafael Romero Carrillo**, con número de matrícula **2015030289**, de la carrera de Ingeniería en **Informática**, ha sido aceptado para realizar su estadía práctica, en esta empresa, durante el período que comprende del **1 de Septiembre al 8 de diciembre**, para cubrir un total de 600 horas.

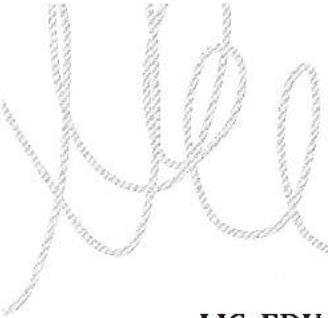
Dicho alumno realizará actividades dentro del departamento, de desarrollo de aplicaciones móviles, bajo la supervisión del **Ing. Marco Antonio Hernández Jiménez, Gerente General.**

Sin otro particular, le envío un cordial saludo.

Atte.



Marco Antonio Hernández Jiménez



Zapopan Jalisco, 12 de Diciembre del 2018

LIC. EDUARDO CRESPO CAMPA.
DIRECTOR DE VINCULACIÓN, DIFUSIÓN Y EXT. UNIVERSITARIA.
UNIVERSIDAD POLITÉCNICA DE SINALOA.

PRESENTE

Por este medio, hago de su conocimiento que el alumno(a) el C. **Mario Rafael Romero Carrillo**, con número de matrícula **2015030289**, de la carrera de Ingeniería en **Informática**, ha cumplido con 600 horas correspondientes a estadía final, en esta empresa/institución, durante el período que comprende del 1 de Septiembre al 8 de Diciembre del año en curso.

Dicho alumno realizó actividades dentro del área/departamento, Desarrollo, bajo la supervisión del **Ing. Marco Antonio Hernández Jiménez, Gerente General.**

Sin otro particular, le envío un cordial saludo.

Atte.



Marco Antonio Hernández Jimenez



UNIVERSIDAD POLITÉCNICA DE SINALOA



C. ROMERO CARRILLO MARIO RAFAEL Folio 2015030289-2018-057
Presente.-

Por medio de la presente me permito comunicarle que **es de aceptarse el tema de tesina**, el cuál se ha solicitado bajo el título:

"Desarrollo de una Página Web para la denuncia ciudadana en faltas al reglamento de tránsito y la educación vial"

mismo que usted desarrollará con objeto de dar lugar a los trámites conducentes para la acreditación de la asignatura de Estadías Profesionales de la Unidad Académica de:

Ingeniería en Informática

Así mismo, le comunico que para el desarrollo de la citada tesina le ha sido asignado como Director Asesor de la misma al: **M.C. Francisco Javier Real Angulo** y como asesores a la **M.S.I Rosa Karina González Trigueros** y a la **M.C. Melissa Osuna Cárdenas**

Sin otro particular por el momento, aprovecho la ocasión para enviarle un cordial saludo.

Atentamente

Dr. Rodolfo Ostos Robles

Director de la Unidad Académica de Ingeniería en Informática
Universidad Politécnica de Sinaloa



SEP





UNIVERSIDAD POLITÉCNICA DE SINALOA



C. ROMERO CARRILLO MARIO RAFAEL
Ingeniero en Informática
Presente.-

Folio: 2015030289-2018-057

Por este conducto le envío un cordial saludo y aprovecho la ocasión para notificarle que el jurado que le fue asignado para evaluar la tesina desarrollada en las estadías profesionales denominada "*Desarrollo de una Página Web para la denuncia ciudadana en faltas al reglamento de tránsito y la educación vial*" y que después de ser revisada en reunión de sinodales ante la Dirección de Ingeniería en Informática, integrado por:

PRESIDENTE DEL JURADO: M.C. Francisco Javier Real Angulo

SINODAL: M.S.I. Rosa Karina González Trigueros

SINODAL: M.C. Melissa Osuna Cárdenas

Ha decidido autorizar y aceptar la digitalización de la misma por el participante, conforme a la normatividad vigente y cumpliendo con los requisitos para tal caso.

Agradeciendo la atención a la presente, le reitero a Usted mi atenta consideración y respeto.

Atentamente

Dr. Rodolfo Ostos Robles
Director de la Unidad Académica de Ingeniería en Informática
Universidad Politécnica de Sinaloa



DIRECCIÓN
INFORMATICA

SEP



AGRADECIMIENTOS

Agradezco a mí y madre por todo su apoyo en lo económico; por asegurarse de que tuviera todo lo necesario para cumplir con mis proyectos universitarios, haciendo algunos sacrificios en el camino, que son en extremo agradecidos.

A mi asesor M.C Francisco Javier Real Angulo por apoyarme en el desarrollo de este trabajo de investigación, que ha prestado su tiempo para que se tuviera el mejor resultado posible y por tenerme la paciencia y comprensión al momento de las entregas en destiempo.

A mi asesor de estadías ING. Domingo Marco Antonio Hernández Jiménez por darme la oportunidad de poder trabajar con él en el desarrollo de este proyecto.

A la Universidad Politécnica de Sinaloa por las múltiples oportunidades que me brindó en mi estancia allí; a mi director de Carrera, Dr. Rodolfo Ostos Robles y a los múltiples profesores que aportaron a mi desarrollo profesional y me ayudaron a llegar a donde estoy ahora.

Contenido

Resumen.....	9
ABSTRACT	9
INTRODUCCIÓN	9
ANTECEDENTES Y PLANTEAMIENTO DEL PROBLEMA.....	11
1.1 Antecedentes	11
1.2 Localización.....	12
1.3 Planteamiento del problema	12
1.4 Propuesta de la investigación.....	13
1.5 Hipótesis	13
1.4 Objetivos	14
1.4.1 Objetivo General.....	14
1.4.2 Objetivos específicos.....	14
1.5 Justificación.....	14
MARCO CONCEPTUAL.....	17
2.1 METODOLOGÍAS	17
2.2 Metodología Scrum	32
2.3 Firebase	35
2.4 Tecnologías web	36
2.5 PHP.....	37
2.6 Mysql.....	38
2.7 Material Design	38
2.8 APIS.....	40
2.9 WinSCP.....	41
2.10 Brackets	43
2.11 Educación vial	44
2.12 jQuery.....	45
2.13 ¿Qué es Javascript?.....	47
DISEÑO METODOLÓGICO	51
3.1 Metodología de la investigación	52
3.2 Desarrollo	54
3.3 Instrumentos.....	55
RESULTADOS DE LA INVÉSTIGACIÓN.....	58

4.1 Resultados	59
CONCLUSIONES Y RECOMENDACIONES	61
BIBLIOGRAFÍA	62
Glosario.....	65

Índice de ilustraciones

Ilustración 1 localización.....	12
Ilustración 2 scrum	33
Ilustración 3 Daily Scrum.....	54
Ilustración 4 Workbench.....	55
Ilustración 5 Ejemplo Pagina web.....	55
Ilustración 6 Braket.....	56
Ilustración 7 Logo firebase	56
Ilustración 8 Ejemplo chat	57
Ilustración 9 Ejemplo página web 2	59
Ilustración 10 Ejemplo página web 3	60
Ilustración 11 Ejemplo página web 4	60

Resumen

El presente documento ha sido realizado por el estudiante Romero Carrillo Mario Rafael con la finalidad de concluir con sus estudios de ingeniería en la Universidad Politécnica de Sinaloa. El trabajo toca el tema de la creación de una página web para ayudar en la educación vial. Con una metodología de desarrollo ágil (Scrum), el proyecto comenzó su creación, para la creación de la plataforma web en la cual se busca que las mismas personas puedan hacer denuncias ciudadanas a faltas en el reglamento vial. Todo esto para ayudar a crear un mejor ambiente en las personas que conducen un auto y los mismos peatones.

ABSTRACT

This document has been made by the student Romero Carrillo Mario Rafael with the purpose of concluding his engineering studies at the Polytechnic University of Sinaloa. The work touches on the creation of a web page to help with road safety education. With a methodology of agile development (Scrum), the project began its creation, for the creation of the web platform in which it is sought that the same people can make citizen complaints to faults in the road regulations. All this to help create a better environment for people who drive a car and pedestrians themselves.

INTRODUCCIÓN

En la actualidad las páginas web son una forma muy destacada de dar a conocer

información que puede ser de mucha relevancia para algunas personas y con la gran cantidad de personas que cuentan con un dispositivo inteligente con acceso a internet hace más fácil su localización.

Una página web, o página electrónica, página digital, o ciberpágina es un documento o información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes y muchas otras cosas, adaptada para la llamada World Wide Web (WWW) y que puede ser accedida mediante un navegador web. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar acceso a otras páginas web mediante enlaces de hipertexto. Frecuentemente también incluyen otros recursos como pueden ser hojas de estilo en cascada, guiones (scripts), imágenes digitales, entre otros.

La educación vial (también conocida como educación para la seguridad vial) es el conjunto de acciones, estrategias, pedagogías, legislaciones y políticas referidas al tránsito y las vías públicas, destinadas a promover la seguridad, prevenir los siniestros, evitar o minimizar los daños y salvar las vidas; a través de la adquisición de conocimientos, destrezas, habilidades, hábitos, valores y actitudes, desde la etapa del preescolar y durante toda la vida.

CAPITULO I

ANTECEDENTES Y PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes

CODING SMART SOLUTIONS

Misión

Proveer de personal calificado, con la experiencia y las habilidades idóneas para realizar labores enfocadas en el ámbito de las Tecnologías de la información, garantizando un resultado satisfactorio gracias a nuestro compromiso, lealtad y equipo de colaboradores altamente capacitados que le aseguran la calidad y el éxito del servicio.

Visión

Ser una empresa líder en el ámbito de las tecnologías de la información, con un enfoque integral como proveedores de personal, de acuerdo a las necesidades de nuestros clientes, con el fin de ser reconocidos a nivel nacional, con presencia en el extranjero y convertirnos en líderes en el ramo, para incrementar el número de clientes con la misma calidad y compromiso que nos distinguen

1.2 Localización

Hacienda México 215 A. Fracc. Haciendas de Aguascalientes, Aguascalientes.



Ilustración 1 localización

1.3 Planteamiento del problema

La Educación vial tiene por objeto desarrollar en el ciudadano; en su condición de

conductor, pasajero o peatón; las aptitudes, destrezas, hábitos y el interés necesario para que disponga de mayor pericia, conocimiento, equilibrio mental; actúe de manera inteligente y razonable; comprenda y respete las leyes, reglamentos y normas vigentes de tránsito y transporte terrestre y así contribuya a prevenir y evitar accidentes viales.

La educación vial se concibe como parte fundamental de la formación y educación ciudadana, en la promoción de una cultura de valoración y respeto por la propia vida y la de los demás. La educación vial cobra mayor relevancia cuando se observa la alta tasa de accidentes de tránsito que se producen diariamente.

De acuerdo con cifras de 2017 de la Organización Mundial de la Salud (OMS), cada año mueren en el mundo cerca de 1,3 millones de personas en accidentes de tránsito, y entre 20 y 50 millones padecen traumatismos no mortales causantes de discapacidad. Los accidentes viales, además, constituyen una de las principales causas de mortalidad en todos los grupos etarios, principalmente entre personas de entre 15 y 19 años. [26]

En México no hay educación vial. No hay una política de Estado encaminada a enseñar de manera formal, en la academia, en la escuela, el principio de respeto al peatón, de dar seguimiento a los reglamentos de tránsito vigentes en el País, ni tampoco de emitir normas que regulen a las escuelas de manejo existentes en la República Mexicana.

1.4 Propuesta de la investigación

Ya con lo antes mencionado se entiende que la educación vial es algo muy importante para poder conducir de manera seguro y que en México es algo se tiene muy poco con esta investigación se busca crear una página web en la cual cualquier ciudadano pueda publicar de manera anónima infracciones que otras personas estén cometiendo para así poder hacer conciencia de lo que se está realizando para poder crear una comunidad que este consiente de lo que hace y se puedan evitar accidentes y se respeten las leyes viales.

1.5 Hipótesis

La implementación de una página web donde los mismos ciudadanos puedan exponer a

personas ignorando las leyes viales puede ayudar mucho ya que los policías de tránsito no pueden estar al tanto de todo lo que sucede en la ciudad y con esto se busca que la mayoría de los ciudadanos hagan conciencia de que se deben respetar las leyes viales.

1.4 Objetivos

1.4.1 Objetivo General

¿Qué beneficios traería a la sociedad el desarrollo de una página web para ayudar en los problemas a la educación vial y reglamento de tránsito?

1.4.2 Objetivos específicos

- Recabar información sobre las principales reglas viales.
- Mostrar los tipos de infracciones viales que se pueden cometer en el contenido de la página.
- Mostrar los lugares en donde se cometen infracciones viales.
- Ayudar a encontrar soluciones con respecto a personas que no estén respetando las leyes de tránsito.
- Hacer públicas a las personas que no respeten las leyes de tránsito.
- Fomentar la educación vial mediante la denuncia ciudadana.

1.5 Justificación

El desarrollo de este proyecto resulta esencial para México ya que es un país con muy poca educación vial donde muy pocos respetan los señalamientos o los lugares para personas discapacitadas y con este proyecto se busca que las mismas personas hagan conciencia de ello dándoles a conocer que lo que están haciendo no está bien y no solo los afecta a ellos sino a las personas a su alrededor. En el país existen más de 38 millones de vehículos automotores registrados según el INEGI eso quiere decir que el 43.5% de los hogares mexicanos disponen de al menos un automóvil por lo cual este

proyecto puede tener un gran alcance en todo México. [27]

Las infracciones de tránsito más comunes en México conforman una larga lista que no parece fácil de reducir teniendo en cuenta que, tan sólo en el Distrito Federal, circulan alrededor de 5 millones de vehículos. Esto, sumado a la escasa educación vial de la sociedad y a la falta de respeto por las normas y por el resto de personas que manejan a nuestro alrededor, así como por el peatón, hace que las infracciones de tránsito ocurran a diario, en muchas ocasiones poniendo en riesgo nuestras vidas y la de los demás. Según la Secretaría de Seguridad Pública del Distrito Federal, cada día se producen unas 500 infracciones de tránsito, dato que refuerza lo comentado anteriormente.

Las infracciones de tránsito más comunes en México son:

- Estacionamiento en zonas prohibidas (impidiendo el acceso a entradas, en doble fila, etc.).
- No usar el cinturón de seguridad.
- Rebasar el límite de velocidad permitido.
- Manejar bajo los efectos del alcohol u otro tipo de drogas, mermando la capacidad física y/o mental para un correcto control del vehículo y del entorno.
- Distracciones al volante por el uso del celular u otro tipo de dispositivos.
- No ceder el paso a los peatones en los cruces destinados para los mismos.
- No respetar las luces de los semáforos ignorando sus indicaciones.
- Manejar sin poseer licencia destinada a este fin. [28]

A demás de que en México hay un alto índice de accidentes viales por no respetar las indicaciones viales y algunas de esas personas nunca son sancionadas por descuido de las autoridades y si alguna persona intenta hacer conocer algún infractor por medio de las redes social en ellas hay mucho contenido y ese intento de hacer conciencia se puede perder por eso se busca un espacio dedicado exclusivamente en este tema. Con la información sobre las principales reglas viales se busca tener contenido que ayude a las personas a informarse sobre las reglas viales para evitar que las cometan en un futuro.

Mostrar los tipos de infracciones viales que se pueden cometer en el contenido de la página para que las personas puedan saber qué tipo de infracciones se pueden cometer

y si alguna persona comente una puedan denunciarla en la página. Al mostrar los lugares donde se comenten las infracciones se busca hacer públicas a las personas que no respetan las leyes viales y mostrar que si es una zona donde se cometen muchas infracciones viales se pueda hablar con las autoridades pertinentes para que se haga algo con la situación y poder llegar a una solución con respecto a las personas que no respeten las reglas de tránsito.

La Cultura vial es un tema que le concierne a toda la sociedad en el mundo. Y principalmente en nuestro país tanto automovilistas como motociclistas, ciclistas y peatones, estamos preocupados con el tema de la escasa cultura vial, pero ¿en realidad hacemos algo al respecto?

CAPITULO II

MARCO CONCEPTUAL

2.1 METODOLOGÍAS

Metodologías

El desarrollo de software, es uno de los sectores tecnológicos más competitivos y no es

algo nuevo, ya que durante muchos años lo ha sido, sin embargo, ha tenido una evolución constante en lo que se refiere a las metodologías o bien, las formas en las cuales se realiza la planeación para el diseño de software, básicamente con el objetivo de mejorar, optimizar procesos y ofrecer una mejor calidad.

Sin embargo, antes de hablar acerca de metodologías y todo este tema tan amplio, analicemos a detalle brevemente ¿Qué es un método? y para que lo acompañemos también veamos ¿qué es una metodología? Seguramente estos términos sonaran familiares, sin embargo, el saber que significan de forma correcta es indispensable.

¿Qué es un Método?

Un Método se compone de diversos aspectos que nos permitirán conseguir una meta o lograr un objetivo. Se define más claramente como un conjunto de herramientas, las cuales, utilizadas mediante las técnicas correctas, permiten la ejecución de procesos que nos llevarán a cumplir los objetivos que buscamos. En pocas palabras y aunque esto lo puedes encontrar como tal en internet, es un conjunto de herramientas, técnicas y procesos que facilitan la obtención de un objetivo.

¿Qué es una Metodología?

En el desarrollo de software, una metodología hace cierto énfasis al entorno en el cual se plantea y estructura el desarrollo de un sistema. Como lo mencioné al principio, existen una gran cantidad de metodologías de la programación que se han utilizado desde los tiempos atrás y que con el paso del tiempo han ido evolucionando.

Esto se debe principalmente a que no todos los sistemas de la información, son compatibles con todas las metodologías, pues el ciclo de vida del software puede ser variable. Por esta razón, es importante que dependiendo del tipo de software que se vaya a desarrollar, se identifique la metodología para el diseño de software idónea.

¿En qué consisten las Metodologías de Desarrollo de Software?

Una Metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Regularmente este tipo de metodología, tienen la necesidad de venir documentadas, para que los programadores que estarán dentro de la planeación del proyecto, comprendan perfectamente la metodología y en algunos casos el ciclo de vida del software que se pretende seguir.

Aunque actualmente existen mucha variedad en metodologías de programación. La realidad es que todas están basadas en ciertos enfoques generalistas que se crearon hace muchos años, algunos tipos de metodologías de desarrollo de software que se utilizaron e inventaron al principio de nuestra era tecnológica y son las que veremos a continuación.

¿Cuáles son modelos del Ciclo de vida del Software tradicionales?

Como les mencioné hace un momento, regularmente, cada metodología de desarrollo de software, tiene un enfoque bien marcado, estos enfoques no son para nada nuevos y se siguen utilizando para la planeación y desarrollo de software aún en nuestros tiempos, así que vamos a ver cuáles son cada uno de ellos y aprenderemos cómo funcionan.

Metodología en cascada: Framework lineal.

El modelo de desarrollo de Software en cascada, es una metodología de la programación muy antigua. Si bien su creador nunca lo menciona como metodología en cascada, el funcionamiento y lineamiento de los procesos de la planeación, son exactamente iguales.

Básicamente, el estilo del modelo en cascada, es que no podrás avanzar a la siguiente fase, si la anterior no se encuentra totalmente terminada, pues no tiene por qué haber vuelta atrás. Vamos a ver cuáles son las fases de desarrollo de software del modelo en cascada, para que te puedas dar una idea.

1. Análisis de Requisitos.

El primer nivel de la modelo cascada, es el análisis de requisitos. Básicamente lo que se documenta aquí, son los objetivos de lo que el software debe hacer al terminar el desarrollo, sin entrar en detalles de la parte interna, los cuales se verán durante el proceso. Sin embargo, es importante señalar que una vez avanzado el paso de análisis, no puede haber vuelta atrás, pues la metodología para el diseño de software con la cual se trabaja no lo permitirá.

2. Diseño del Sistema.

Todo lo que conlleva el armado de un diseño para el sistema que vayas a utilizar, es lo que continua después del análisis de requisitos. Aquí se elaborará lo que es la estructura del sistema y se determinarán las especificaciones para cada una de las

partes del sistema que se planea desarrollar. Siendo del mismo modo, una fase a la cual ya no se podrá volver después de haber bajado al nivel 3.

3. Diseño del Programa.

En este punto, aún no ingresamos a lo que es la escritura de código, sin embargo, ya se realizan los algoritmos que se van a utilizar en la programación. Si bien recuerdas, un algoritmo no necesariamente es código, simplemente tomas una hoja de papel y escribes el algoritmo que vas a utilizar. Esto es precisamente lo que se realiza en el paso número 3.

4. Codificación.

Seguramente como programador, esta es la parte que estabas esperando, pues ahora es momento de empezar a escribir todo el código que será necesario para el desarrollo del software. Para este punto, la velocidad y el tiempo que se requiera, dependerá mucho del lenguaje de programación que vayas a utilizar. Pues algunos lenguajes de programación permiten utilizar componente, bibliotecas e incluso algunas funciones para reutilizar código, las cuales podrán acelerar el proceso de codificación en gran manera.

5. Ejecución de Pruebas.

La codificación ha terminado y ahora es momento de verificar que nuestro sistema es realmente funciona, antes de que el cliente empiece a utilizarlo. Este es precisamente el objetivo de la fase 5 de pruebas. Aquí es recomendable que intentes mover lo más que se pueda tu software, con el objetivo de dañarlo intencionalmente, de este modo, si supera las pruebas de daño realizadas por ti, entonces estará listo para el usuario final.

6. Verificación.

Después de haber realizado una gran cantidad de pruebas en la Fase 5, debemos migrar a la verificación. Esta fase consiste en la ejecución del Software por parte del usuario final. Si la fase cinco se realizó correcta y profundamente, el software no tendrá ningún tipo de problema y el usuario final quedará satisfecho con el resultado.

7. Mantenimiento.

Seguramente te has dado cuenta, de que las aplicaciones o el software actual, constantemente se está actualizando. Esto se debe principalmente a que se le da mantenimiento al software, se solucionan errores, se quitan algunos bugs, se añaden

funcionalidades, todo después de que el usuario final ya lo ha probado y utilizado en su fase final. Esta es posiblemente una de las fases más tediosas del modelo de desarrollo de software, pues debes estar atento a los comentarios de los usuarios, para ver qué cosas son las que no funcionan correctamente y a las cuales hay que darles mantenimiento.

¿Cuáles son los Principios básicos del modelo de cascada?

Como puedes ver, el proceso de desarrollo de software con el modelo de cascada es bastante complejo. Sin embargo, uno de sus principios es que cada una de las fases elaboradas, se encuentre documentada perfectamente, de este modo, si el desarrollo queda suspendido en alguna fase, cualquier usuario que quiera continuar con el proyecto lo podrá hacer leyendo la documentación.

Así también, es muy común encontrar metodologías para el desarrollo de software en cascada con fechas de objetivos, tiempos o presupuestos para determinadas fases. Aprovechando el hecho de que una vez que avanzaste de fase, es muy poco recomendable el volver atrás, aunque regularmente se tiene un cierto nivel de tolerancia, pero lo correcto en la utilización del modelo de cascada, es que no puedas ir atrás a realizar modificaciones de ningún tipo.

Método de Prototipos

Esta metodología de la programación todavía sigue siendo la favorita de muchos. Consiste básicamente en que en base a los requerimientos y necesidades que tiene el cliente, se realiza de forma rápida un prototipo, este no vendrá completo ni mucho menos terminado, pero si permitirá contar con las bases necesarias para que cualquier programador pueda seguir trabajando en el hasta llegar al código final.

Por si no lo sabes aún, un prototipo es una versión no terminada del producto que se le entregará al cliente o usuario final. Esto nos genera cierta ventaja en el desarrollo de productos similares con funciones distintas, por ejemplo. Supongamos que vas a desarrollar un proyecto para 3 clientes distintos, ambos con una estructura idéntica, pero con funcionalidades muy distintas, entonces lo que hacemos es crear un prototipo base y entorno al mostrarlo a nuestros clientes para que de ahí se empiecen a desarrollar las demás funciones.

Mejor vamos a ver cuáles son las etapas de desarrollo de software por las cuales

tendrás que pasar, en caso de utilizar la metodología de prototipos.

1. Planeación.

A diferencia de otras metodologías, la planeación debe ser muy rápida, en esta fase no puedes demorarte mucho, pues recuerda que solamente será un prototipo por el momento.

2. Modelado.

Nuevamente, una fase que deberá ser suficientemente rápida como para que nos quite nada de tiempo. Hacer el modelado será simple y te sigo recordando que solamente es un prototipo, al menos por ahora.

3. Elaboración del Prototipo.

Ya que contamos con la planeación de lo que vamos a realizar y el modelado rápido, entonces es momento de elaborar el prototipo. Para esta instancia, ya no te diré que lo debes hacer rápido, puesto que te tomará el tiempo que tenga sea necesario elaborarlo, recuerda que este ya se muestra al cliente, así que ya es una fase importante.

4. Desarrollo.

Posterior a contar con el prototipo elaborado y mostrado al cliente, es momento de comenzar el desarrollo. Este te tomará una gran cantidad de tiempo, dependiendo del tamaño del proyecto y el lenguaje de programación que se vaya a utilizar.

5. Entrega y Retroalimentación.

Una de las cosas con las que cuenta el modelo de prototipos, es que, una vez entregado el proyecto, debemos darle al cliente cierta retroalimentación sobre cómo utilizarlo y ciertamente es una fase que se encuentra dentro de las etapas de desarrollo de software esta metodología.

6. Comunicación con el Cliente.

Es importante que, una vez entregado el proyecto, tengamos cierta comunicación con el cliente, básicamente para que nos indique si el proyecto es correcto o si desea agregarle ciertas funciones, nuestra metodología lo permite. Si fuera en modo cascada, entonces sería algo realmente imposible de hacer.

7. Entrega del Producto Final.

Por último, solamente quedará entregar el sistema elaborado mediante esta metodología. Aquí tendrás la ventaja de que el código es reutilizable, para que así con

el prototipo ya puedas simplemente empezar de nuevo y con una buena base de código que te acelerará el proceso.

¿Cuáles son los Principios Básicos del método de prototipos?

Por supuesto, te habrás dado cuenta de que el modelo de prototipos puede llegar a ser un poco más tedioso, aunque todo dependerá del ámbito en que lo utilices. Sin embargo, uno de sus principios básicos que seguramente habrás notado, es que con el método de prototipos el proyecto se va dividiendo en partes cada vez más pequeñas, para evitar el peligro ante los riesgos frente a los que estamos expuestos.

Además, otros de sus principios básicos fundamentales, es que, con la metodología de prototipos, el cliente final se involucra mucho más en el proyecto que con otras metodologías, haciendo de esta forma que el producto final llegue rápidamente, aunque con un poco más de presión en el proceso. La ventaja es que conforme se van haciendo prototipos pequeños, poco a poco se va llegando al producto final. Incluso en algún determinado momento podrás llegar a crear un prototipo que, con solo ajustar ciertos detalles, se podría convertir en el producto que el usuario quiere.

Modelo Incremental o Iterativo y Creciente

El modelo Incremental, es una metodología de la programación muy utilizada hoy en día, pues su comodidad de desarrollo permite que te obtenga un producto final mucho más completo y exitoso. Se trata especialmente de la combinación de los modelos lineal e iterativo o bien, modelo de cascada y prototipos. Básicamente consiste en completar varias iteraciones de lo que es el modelo de cascada, pero sin completar ninguna, haciendo iteraciones lo que se hace es crear una evolución en el producto, permitiendo que se agreguen nuevas especificaciones, funcionalidades, opciones, funciones y lo que el usuario requiera después de cada iteración. **[10]**

En pocas palabras, el Modelo Incremental repite el modelo de cascada una y otra vez, pero con pequeñas modificaciones o actualizaciones que se le puedan ir agregando al sistema. De este modo el usuario final se ve sumamente sumergido en el desarrollo y puedes proporcionarle un resultado óptimo.

Fases del Modelo Incremental

El modelo iterativo o incremental, cuenta con algunas fases de desarrollo de software que realmente no tienen mucha complejidad, vamos a verlas:

1. Inicialización.

como en todo modelo de desarrollo, debe haber una inicialización, aquí se puede hablar de dar una idea, de tener algunos requisitos que se buscan en el proyecto y ciertas especificaciones que se pueden manejar. No es necesario que se haga una lista total de requerimientos pues recordemos que el proyecto se basa en iteraciones, así que el proyecto puede ir evolucionando poco a poco.

2. Periodos de Iteración.

Durante el desarrollo del proyecto, es cuando damos inicio a las iteraciones. La primera iteración se realiza con las características iniciales, básicamente al final de la primera iteración, queda un pequeño prototipo de lo que será el proyecto, pero se puede volver a inicializar la iteración y realizar modificaciones en los procesos, como el análisis y las especificaciones o funciones que el usuario final requiere para su sistema. El número de iteraciones que se realicen son ilimitadas y dependerá tanto del desarrollador como del usuario final. Si nuestro objetivo es que el cliente o la persona que necesita el trabajo que de completamente satisfecha, entonces nos veremos en la necesidad de hacer la cantidad de iteraciones que se requieran al final del día.

3. Lista de Control.

Es importante que conforme se vaya realizando cada iteración, se vaya llevando un control del mismo en una lista. Como si fuera un programa que recibe actualizaciones constantemente. Cada una de las actualizaciones o iteraciones deberá ser documentada y de ser posible, guardada en sus respectivas versiones, para que sea sencillo volver atrás, en caso de que una iteración no sea exitosa o el usuario ya no la requiera.

¿Cuáles son los principios básicos del Modelo Incremental?

Es importante definir los siguientes principios fundamentales, pues nos permiten saber con claridad por dónde va la idea de la metodología.

La idea de un modelo incremental, es utilizar una serie de mini modelos de desarrollo de software en cascada, segmentando requerimientos y permitiendo que el usuario vaya de la mano con el proyecto durante la realización.

Básicamente las fases de cada iteración, son las mismas que se manejan en el modelo de cascada, aunque también se pueden agregar algunas, pero su objetivo es completar

cada fase de la iteración, para que esta se vaya complementando poco a poco y no se genere un desarrollo tedioso y cansado que puede alargar la duración del proyecto en demasía.

Debes saber, que cada iteración te generará un prototipo cada vez más evolucionado, estos deberás guardarlos por si en determinado momento deseas volver atrás, pues a diferencia del modelo de cascada, podemos retroceder cuando se requiera y los prototipos se pueden volver a utilizar una y otra vez, pues el código fuente es reutilizable.

Modelo en Espiral

El modelo en espiral, fue utilizado y diseñado por primera vez por Barry Boehm en 1986. Se trata nuevamente de una combinación entre el modelo lineal o de cascada y el modelo iterativo o basado en prototipos, sin embargo, a este sistema lo que debemos añadirle es la gestión de riesgos, algo que en los modelos anteriores ni siquiera se menciona.

Este modelo, consiste en ciertas fases que se van realizando en modo de espiral, utilizando procesos de la misma forma en que se utilizan en el modelo de cascada, sin embargo, aquí estos no son obligatorios y no llevan precisamente el orden establecido. Básicamente se trata de un modelo evolutivo, que conforme avancen los ciclos, irá incrementando el nivel de código fuente desarrollada, un incremento en la gestión de riesgos y por supuesto un incremento en los tiempos de ejecución y planificación del sistema, esto es lo que tiene el modelo en espiral.

Para que tengas una idea más clara, el modelo en espiral es principalmente utilizado para el desarrollo de grandes proyectos como la creación de un sistema operativo. Sin embargo, necesitas de ciertos requisitos, como el hecho de contar con personal completamente capacitado para las funciones que se requieran. Mejor veamos cuales son las fases o tareas dentro del modelo de espiral.

1. Determinar Objetivo.

Es importante que siempre consideres una planeación inicial, esta solo se realizará una vez. Sin embargo, el proceso de determinar objetivos se hará constantemente durante cada iteración que se vaya realizando con el modelo espiral. Esto se debe a que poco a poco se irá incrementando más el tamaño del manual de usuario, los requisitos, las

especificaciones e incluso las restricciones. Todo esto entra en lo que es la tarea de objetivos y con cada vuelta en el espiral entraremos a esta tarea, la cual como todas las demás, es fundamental.

2. Análisis de Riesgo.

Una etapa donde incluso una lluvia de ideas podría ayudar, el análisis de riesgos. Aquí deberás tener en cuenta todo aquello que pueda dañar a tu proyecto, ya sea que se trate de ciertas amenazas o de posibles daños que se puedan ocasionar, teniendo además un Plan B por así decirlo, para que en caso de que ocurra algo inesperado, tener a la mano la solución para continuar con el proyecto.

En esta fase del modelo espiral, podemos agregar lo que son la creación de prototipos, pues siempre es bueno tener un respaldo de nuestro código, se esta forma en caso de que algo malo suceda, volvemos a la versión anterior. Así que cada vez que vayamos a ingresar a la fase de pruebas e implementación, será necesario contar con un prototipo que nos respalde.

3. Desarrollar, Validar y Probar.

Básicamente en esta fase, la forma en que se estará desarrollando el proyecto, dependerá del análisis de riesgos, pues siempre se va a ir desarrollando el proyecto enfocándose en los riesgos que podemos evitar en nuestro software, es decir, si la situación de riesgo más obvia se encuentra en la interfaz del usuario, entonces hay que trabajar con prototipos para este enfoque, creando evoluciones proporcionales, para ir evitando ese tipo de riesgos. Esto no significa que ignoremos el resto del proyecto o del desarrollo, sin embargo, el modelo en espiral si acomoda un poco más las prioridades al momento, independientemente de todo lo demás. Por lo que siempre en cada vuelta o iteración que se le dé al modelo de espiral, tu avance siempre dependerá del análisis de riesgo, hasta que este sea mínimo y el desarrollo pueda continuar de forma normal.

4. Planificación.

Antes de proceder a realizar otra iteración o vuelta a la espiral, debemos prestar atención a lo que sucedió en la vuelta anterior. Debemos analizar detalladamente si los riesgos encontrados ya tuvieron solución, lo cual debe ser lo ideal, puesto que ahora habría que analizar más especificaciones y más requisitos del sistema para continuar con el desarrollo. Básicamente la fase de planificación, nos servirá para determinar el

avance de nuestro proyecto e indicar hacia dónde vamos a dirigirnos en la próxima iteración.

¿Cuáles son los Principios básicos del modelo en Espiral?

Está claro que el modelo en espiral, es sumamente distinto a los demás. Encontramos por fuera cuatro fases bien organizadas, las cuáles siempre deben llevar ese orden que se estipula desde el principio. Una determinación de objetivos, un análisis de riesgos, el desarrollo y las pruebas, junto con la planificación, la cual dependerá de los resultados de la iteración para saber cómo se actuará en la siguiente vuelta al espiral.

Básicamente, en el modelo de espiral, toda la atención está enfocada hacia el análisis de riesgos, pues el objetivo primario será reducir los riesgos que se vayan generando, de otra forma el sistema no llegará a ser seguro jamás.

Algo muy importante que debes ver en el modelo de espiral, es que los interesados deben estar involucrados prácticamente en cada vuelta que se dé al espiral, para crear lo que son los requisitos antes de realizar una vuelta más y al final en la fase de planificación, se determinan los logros obtenidos, el avance y lo que se esperará de una siguiente vuelta.

Metodología RAD

A diferencia de otras metodologías para el desarrollo de software, la metodología RAD o desarrollo rápido de aplicaciones, no cuenta con una serie de fases ordenadas por así decirlo. Aunque si está basada en lo que es el modelo de cascada y la creación de prototipos, sin embargo, el proceso es muy independiente a contar con ciertas fases estipuladas como los modelos que hemos visto anteriormente. Así que vamos a ver los principios del modelo RAD.

¿Cuáles son los principios básicos del Modelo RAD?

Del mismo modo que modelos anteriores, el Modelo RAD, está basado en el uso de las iteraciones y principalmente en el manejo de prototipos. Sin embargo, a diferencia del resto, la metodología RAD hace uso de las Herramientas CASE, las cuales permitirán acelerar el proceso considerablemente.

Del mismo modo que el modelo espiral y el de prototipos, RAD se subdivide en pequeñas secciones, las cuales se irán optimizando y de esta forma se irá avanzando mucho más rápido que con grandes segmentos que pueden volverse difíciles dentro de

un proceso acelerado como lo esté este modelo.

Una de las ventajas del RAD, es que el enfoque y las prioridades van hacia la fase de desarrollo, a diferencia de modelos como el espiral, que se enfoca en que los riesgos al momento sean mucho menores. Acá con el RAD, se hace lo contrario, si hay riesgos reducimos los requerimientos para reducir los riesgos, no como en el espiral, que, entre más riesgos, más requisitos aportamos para que se incremente. Acá la idea es reducir tiempos y no riesgos, o si, talvez también, pero la reducción de riesgos es totalmente inversa al modelo espiral.

Por supuesto, como en todos los modelos, vas a requerir de ciertos factores documentados, de preferencia todo lo que se pueda, para que en caso de que alguien más venga a continuar con este proyecto, tenga a la mano toda la información que necesita y con unas cuentas lecturas pueda empezar a desarrollar lo que se había quedado pendiente en un determinado momento. [22]

Metodologías ágiles

Hablar de metodologías ágiles implica hacer referencia a las metodologías de desarrollo de software tradicionales ya que las primeras surgieron como una reacción a las segundas; sus características principales son antagónicas y su uso ideal aplica en contextos diferentes.

- Metodologías tradicionales

Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de licitación de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario.

En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta.

- Metodologías Ágiles

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la

realidad de cada equipo y proyecto.

Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características.

Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas. La comunicación con el cliente es constante al punto de requerir un representante de él durante el desarrollo. Los proyectos son altamente colaborativos y se adaptan mejor a los cambios; de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente. [1]

- Comparación entre metodologías

La tabla 1 muestra aspectos relevantes de las metodologías de desarrollo tradicional contrastándolas con los aspectos relevantes de las metodologías de desarrollo ágil.

Tabla 1. Metodologías tradiciones vs metodologías Ágiles

Metodologías tradicionales	Metodologías ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

- Manifiesto por el desarrollo Ágil

Las metodologías ágiles son flexibles, sus proyectos son subdivididos en proyectos más pequeños, incluyen comunicación constante con el cliente, son altamente colaborativos y se adaptan mejor a los cambios. De hecho, el cambio en los requerimientos es una

característica esperada al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente.

En 2001 se crea el Manifiesto por el desarrollo ágil de software, documento en el que se acuerdan cuatro principios básicos para el desarrollo de software, que establece prioridades y marca diferencias de fondo frente a los sistemas tradicionales: individuos e interacciones, por encima de procesos y herramientas.

Revisión de metodologías ágiles para el desarrollo de software de documentación extensiva; colaboración con el cliente, por encima de negociación contractual; y respuesta ante el cambio, por encima de seguir un plan.

Los principios que dan origen al manifiesto implican la satisfacción del cliente mediante entregas tempranas y continuas de software que funcione; requerimientos cambiantes en cualquier etapa del proyecto; participación activa del cliente; simplicidad; equipos de desarrollo motivados y auto-organizados; comunicación efectiva; auto inspecciones; y adaptación.

El manifiesto por el desarrollo ágil de software es el resultado del trabajo colaborativo de un grupo formado por diecisiete personas, entre desarrolladores de software, escritores y consultores, quienes lo construyeron y suscribieron en 2001. La firma y publicación del Manifiesto en ese año no implica que esa sea la fecha de origen de las metodologías ágiles o que antes de ese año no existieran, sino el reconocimiento de la necesidad –y la expresión– de un lineamiento común capaz de hacer posible algún tipo de agrupación entre ellas. [2]

Las metodologías ágiles se caracterizan por el desarrollo iterativo e incremental; la simplicidad de la implementación; las entregas frecuentes; la priorización de los requerimientos o características a desarrollar a cargo del cliente; y la cooperación entre desarrolladores y clientes. Las metodologías ágiles dan como un hecho que los requerimientos van a cambiar durante el proceso de desarrollo. [3]

- SELECCIÓN DE METODOLOGÍAS

En la revisión de las metodologías ágiles para el propósito de seleccionar aquellas en las que se enfocará este artículo se sigue una metodología simple, propuesta por los autores, y que se describe a continuación.

Una primera selección surge del manifiesto: Scrum, Extreme Programming [XP], Dynamic SystemDevelopment Method [DSDM], Crystal, Adaptive Software Development [ASD] y Feature-Driven Development [FDD], están representadas en él, a través de al menos una de las personas que lo suscribieron. [1]

Otra aproximación para definir las metodologías ágiles más relevantes consiste en revisar las que han sido citadas y explicadas en libros de ingeniería de software. Al respecto, Sommerville indica que las metodologías ágiles más conocidas son XP, Scrum, Crystal, ASD, DSDM y FDD, mientras que Pressman [4] centra su trabajo en la explicación del funcionamiento de XP, DSDM, ASD, Scrum, FDD y Agile Modelling [AM]. Una revisión de las investigaciones, revisiones e implementaciones referenciadas por asociaciones como IEEE y ACM, arroja un tercer parámetro [5]. Las metodologías más populares en los documentos científicos entre el 2003 y el 2007 fueron XP y Scrum. Sin embargo, cabe aclarar que la mayoría de lo que se publicó no está relacionado con una metodología en particular, sino con el concepto de metodologías ágiles en general; de hecho, el eje común a los temas tratados con mayor frecuencia es la presentación de experiencias exitosas de implantación. [6]

Otro criterio de selección es el reconocimiento de la alta adopción en la industria de desarrollo de software en los últimos años. Según la séptima versión de la encuesta anual del estado del desarrollo de software usando metodologías ágiles, el tamaño promedio de las organizaciones es de 100 empleados. De quienes contestaron la encuesta, el 84% dijo que en sus compañías se practican las metodologías ágiles, lo que en algunos casos se ha venido haciendo por más de 5 años.

La tendencia de definir a XP y a Scrum como los métodos de desarrollo ágil más relevantes es notoria, así se evidencia en los trabajos de Jiang y Eberlein [7], Thorstein, Hannay, Pfahl, Benestad, y Langtangen [8], , Silva, Selbach, Maurer, y Hellmann [9], Maurer y Hellmann , Vijayasathy y Turk , Hoda, Kruchten, Noble, y Marshal y Szöke,[12] así como en encuestas hechas por importantes proveedores de software para la gestión de este tipo de metodologías. Thorstein et ál incluso listan todas las prácticas ágiles existentes, tomando en cuenta solamente XP y Scrum. [10,11]

La mayoría de equipos ágiles exitosos han adaptado prácticas ágiles de distintas metodologías para generar un proceso de desarrollo propio que se ajusta a sus

necesidades. Estas adaptaciones parecen estar centradas en mezclas de Scrum con XP. XP se enfoca en prácticas de desarrollo mientras que Scrum apunta a la administración de proyectos.

2.2 Metodología Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.



Ilustración 2 scrum

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente (Product Owner) prioriza los objetivos balanceando el valor que le aportan respecto a su coste (que el equipo estima considerando la Definición de Hecho) y quedan repartidos en iteraciones y entregas.

Las actividades que se llevan a cabo en Scrum son las siguientes (*los tiempos indicados son para iteraciones de 2 semanas*):

Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1. Selección de requisitos (2 horas). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que prevé que podrá completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
2. Planificación de la iteración (2 horas). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas, se autoorganizan para trabajar incluso en parejas (o grupos mayores) con el fin de compartir conocimiento (creando un equipo más resiliente) o para resolver juntos objetivos especialmente complejos.

Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 minutos), normalmente delante de un tablero físico o pizarra (Scrum Taskboard). El equipo inspecciona el

trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con la previsión de objetivos a mostrar al final de la iteración. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
- ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
- ¿Qué impedimentos tengo o voy a tener que nos impidan conseguir nuestro objetivo?

Durante la iteración el Facilitador (Scrum Master) se encarga de que el equipo pueda mantener el foco para cumplir con sus objetivos.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar el objetivo de la iteración o su productividad.

Durante la iteración, el cliente junto con el equipo refinan la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o replanifican los objetivos del proyecto (10%-15% del tiempo de la iteración) con el objetivo de maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

1. Revisión (demostración) (1,5 horas). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
2. Retrospectiva (1,5 horas). El equipo analiza cómo ha sido su manera de trabajar

y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de eliminar o escalar los obstáculos identificados que estén más allá del ámbito de acción del equipo. [13]

2.3 Firebase

Firebase es la nueva y mejorada plataforma de desarrollo móvil en la nube de Google. Se trata de una plataforma disponible para diferentes plataformas (Android, iOS, web), con lo que de esta forma presentan una alternativa seria a otras opciones para ahorro de tiempo en el desarrollo como Xamarin.

¿En qué consiste Firebase? Es la evolución de una plataforma que ha ido mejorando desde que Google la compró en 2014 y luego la continuó mejorando con la compra del equipo de Divshot. Veamos en detalle porque tiene todos los ingredientes para convencer a propios y extraños.

Firebase, una plataforma en la nube para todos

En muchas ocasiones nos planteamos cómo poder acceder a un servicio web para poder tener nuestra aplicación trabajando con datos en la nube. Por ello surgió Firebase, para proveer una API para guardar y sincronizar datos en la nube en tiempo real.

Sus características fundamentales están divididas en varios grupos, las cuales podemos agrupar en:

- **Análíticas:** Provee una solución gratuita para tener todo tipo de medidas (hasta 500 tipos de eventos), para gestionarlo todo desde un único panel.
- **Desarrollo:** Permite construir mejores apps, permitiendo delegar determinadas operaciones en Firebase, para poder ahorrar tiempo, evitar bugs y obtener un aceptable nivel de calidad. Entre sus características destacan el almacenamiento, testeo, configuración remota, mensajería en la nube o autenticación, entre otras.
- **Crecimiento:** Permite gestionar los usuarios de las aplicaciones, pudiendo además captar nuevos. Para ello dispondremos de funcionalidades como las de invitaciones, indexación o notificaciones.

- Monetización: Permite ganar dinero gracias a AdMob.

La documentación de Firebase, una delicia Google no ha obviado ningún detalle, y a la hora de gestionar cualquiera de los módulos anteriores, tenemos disponible una gran cantidad de información, así como de ejemplos o incluso tutoriales.

Todo ello, tanto para iOS como para Android como para web:

En esta página podremos encontrar toda la información necesaria, clasificada por las categorías que anteriormente os citábamos, incluyendo cada una de las plataformas, de modo que todo será muy intuitivo para leer, y nos resultará muy sencillo su uso gracias a los ejemplos con código real.

En esta vida todo tiene un precio

Pero la parte negativa de Firebase es que no es totalmente gratis. Es decir, disponemos de 3 planes según nuestras necesidades:

Como observareis, un plan es gratuito, pero presenta una serie de limitaciones en espacio almacenado y conexiones simultáneas, que quizá no en todos los casos será suficiente para el desarrollador. En estos casos tendremos que pasarnos a otra de las tarifas, entre las que tenemos una tarifa plana de 25 dólares al mes (con límite también), u otra que ya va en función del consumo que tengamos.

Pero a pesar de ello, para todos los que necesitéis hacer uso de la nube es una de las mejores opciones, pues además conseguiréis delegar en Firebase una gran cantidad del desarrollo necesario, ahorrando tiempo (y por tanto, dinero). [14]

2.4 Tecnologías web

Las tecnologías Web sirven para acceder a los recursos de conocimiento disponibles en Internet o en las intranets utilizando un navegador. Están muy extendidas por muchas razones: facilitan el desarrollo de sistemas de Gestión del Conocimiento (en lo adelante GC), su flexibilidad en términos de escalabilidad, es decir, a la hora de expandir el sistema; su sencillez de uso y que imitan la forma de relacionarse de las personas, al poner a disposición de todos los conocimientos de los demás, por encima de jerarquías, barreras formales u otras cuestiones. Estas tecnologías pueden llegar a proporcionar recursos estratégicos, pero, evidentemente, no por la tecnología en sí misma, que está disponible ampliamente, sino por lo fácil que es personalizarla y construir con ella

sistemas de GC propietarios de la empresa.

Internet, Intranet o extranet permiten a los usuarios el acceso a una gran cantidad de información: leer publicaciones periódicas, buscar referencias en bibliotecas, realizar paseos virtuales por museos, compras electrónicas y otras muchas funciones. Gracias a la forma en que está organizada la World Wide Web (WWW), los usuarios pueden saltar de un recurso a otro con facilidad. [15]

2.5 PHP

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation 2.0 considera esta licencia como software libre. PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad, robustez y modularidad.

Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML. En este documento no se hará especial énfasis en explicaciones en detalle de la sintaxis y el funcionamiento del código dentro del sistema. Comparado con otros lenguajes, como ASP la principal ventaja de PHP es su carácter multiplataforma. Por otro lado, los programas en ASP resultan más lentos y pesados, y también menos estables.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Es también el módulo Apache más popular, el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente.

Mediante extensiones es también posible la generación de archivos PDF, Flash, así

como imágenes en diferentes formatos [3]. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI. [16]

2.6 Mysql

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2. MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos.

Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, incluyendo algunos de los que probablemente no ha oído nunca hablar. MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de relacionales, que presentaremos en una sección posterior. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos. [17]

2.7 Material Design

Es una nueva filosofía enfocada, inicialmente, para el diseño Android; sin embargo, su implementación en el mundo del desarrollo ha sido tal, que muy pronto comenzó a extenderse a toda la web. Es así como la encontramos, actualmente, no solo en aplicaciones Android, sino en páginas web y demás plataformas de software.

El chileno Matías Duarte, diseñador de interfaces informáticas de usuario y actual vicepresidente de diseño de Google, es el creador de este concepto de diseño que

revolucionó el aspecto visual y experiencial de los programas desarrollados en la actualidad.

¿En qué se basa?

Su nombre traduce “Diseño material”, y... ¿Qué significa? Esta filosofía de diseño basó su nombre en los objetos materiales. Según la definición oficial de Google: “es un lenguaje que combina los principios innovadores de la tecnología con las normas clásicas del diseño”.

Se debe, entonces, a una serie de pautas en las cuales los colores, las sombras, la profundidad, las superficies y los bordes juegan un papel protagonista, tal como en la realidad material que experimentamos a diario.

Matías Duarte, creador de este concepto, explicó que “a diferencia del papel, el Diseño Material se puede ampliar y redimensionar de manera inteligente. Material Design tiene superficies físicas y bordes. Las escenas y sombras proporcionan significado sobre lo que se puede tocar y cómo se va a mover”

Material Design se integró en la versión 5.0 (Android Lollipop) como reemplazo al ampliamente conocido estilo de diseño Holo (cuyo nombre se debe a su similitud con la apariencia de un holograma).

¿Cuáles son sus características?

Desde su presentación en la pasada conferencia Google I/O 2014, el concepto de Material Design adquirió gran importancia en el mundo de los diseñadores digitales, quienes comenzaron a implementar sus características visuales en los proyectos que a diario realizaban. Veamos cuáles son estas características:

- En Material Design, la interacción de los objetos en el espacio es transmitida a través de los principios de la luz, la superficie y el movimiento.
- Se trata del espacio en 3D, en el cual los objetos tienen tres dimensiones: la anchura, la altura y el grosor.
- Es influenciado por los elementos clave de diseño de impresión, es decir, fuentes, colores, imágenes, grids, escalas y espacio, los cuales crean una estructura jerárquica y significativa que da protagonismo a la inmersión del usuario en la experiencia digital.

- Los objetos en el entorno de Material Design tienen las relaciones padre-hijo que determinan su interacción. Cada objeto puede estar subordinado a un solo objeto principal o padre y puede tener cualquier número de hijos. Los objetos tienen todas las propiedades de su “padre”, por ejemplo, posición, rotación y elevación.

2.8 APIS

Una API (siglas de ‘Application Programming Interface’) es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

Las API pueden servir para comunicarse con el sistema operativo (WinAPI), con bases de datos (DBMS) o con protocolos de comunicaciones (Jabber/XMPP). En los últimos años, por supuesto, se han sumado múltiples redes sociales (Twitter, Facebook, Youtube, Flickr, LinkedIn, etc) y otras plataformas online (Google Maps, WordPress...), lo que ha convertido el social media marketing es algo más sencillo, más rastreable y, por tanto, más rentable.

Las API son valiosas, ante todo, porque permiten hacer uso de funciones ya existentes en otro software (o de la infraestructura ya existente en otras plataformas) para no estar reinventando la rueda constantemente, reutilizando así código que se sabe que está probado y que funciona correctamente. En el caso de herramientas propietarias (es decir, que no sean de código abierto), son un modo de hacer saber a los programadores de otras aplicaciones cómo incorporar una funcionalidad concreta sin por ello tener que proporcionar información acerca de cómo se realiza internamente el proceso.

Ejemplos de uso de las API web

Los desarrolladores de un programa cualquiera para Windows que se conecte a Internet no necesitan incluir en su código las funciones necesarias para reconocer la tarjeta de red, por ejemplo: basta una ‘llamada’ a la API correspondiente del sistema operativo.

Las plataformas relacionadas con Twitter, sean las de búsqueda y filtrado de menciones como Topsy, o las de gestión de tuits como Hootsuite, pueden hacer uso de la capacidad de procesamiento de los servidores de Twitter.

Los webmasters pueden incluir en sus webs de forma automática productos actualizados de Amazon o eBay, permitiendo iniciar el proceso de compra desde su

web. O quizá te permiten identificarte con Facebook Connect, ahorrándose así el complejo proceso de gestión de usuarios (y evitando al internauta tener que registrarse en una web más). Igualmente, los botones de “+1” de los blogs son llamadas a la API de Google.

En definitiva, podemos ver el uso de las API como una ‘subcontratación’ de funciones. Por supuesto, esta subcontratación impone condiciones al subcontratante: algunos sitios como Twitter o eBay limitan el número de llamadas que un determinado software o web pueden hacer a su API en un determinado período de tiempo (por minuto, hora o día) antes de tener que pagar una licencia. [18]

2.9 WinSCP

WinSCP es una aplicación de Software Libre. WinSCP es un cliente SFTP gráfico para Windows que emplea SSH. También se puede seguir usando la versión anterior del protocolo. Su función principal es facilitar la transferencia segura de archivos entre dos sistemas informáticos, el local y uno remoto que ofrezca servicios SSH.

Esta página es una pequeña introducción en castellano, ya que la mayor parte de la documentación de WinSCP se encuentra únicamente en inglés, así como la mayor parte del contenido referido en los enlaces.

Características

- Interfaz gráfica (GUI).
- Disponible en varios idiomas.
- Integración con Windows (drag&drop, URL, accesos directos)
- Soporte de las operaciones comunes de archivo.
- Soporte de protocolos SCP y SFTP sobre SSH-1 y SSH-2
- Soporte de operaciones programadas (batch), guiones e interfaz de línea de comandos.
- Sincronización de directorios de varias maneras, semi o completamente automatizadas.
- Editor de texto integrado
- Soporte de autenticación mediante contraseñas SSH, método keyboard-interactive, clave pública o Kerberos (GSS)

- Se integra con Pageant (Agente SSH de PuTTY) para ofrecer soporte completo de autenticación mediante clave pública
- Interfaces similares al Explorador de Windows (panel único) o al Comandante Norton (panel dual).
- Opcionalmente es posible guardar la información de sesión.
- Posibilidad de almacenar la configuración del programa en un archivo de configuración en vez de en el registro de Windows, lo que facilita su uso desde unidades portátiles, como discos USB y GVC.

Operaciones de archivo

WinSCP permite efectuar las operaciones básicas con archivos, tales como descargas y subidas. También es posible renombrar archivos y directorios, crear nuevos directorios, modificar las propiedades de archivos y carpetas, y crear enlaces simbólicos y accesos directos.

Mediante el uso de paneles duales es posible también administrar archivos en el equipo local. Recordar que es gratuito.

Es fácil de usar para aquellos usuarios que no usan normalmente la Consola de cualquier terminal Linux, desde Windows. Al punto de copiar, mover o eliminar archivos sin problema y rapidez.

Conexión con equipos remotos

WinSCP permite conectarse a un servidor SSH (Secure Shell) empleando el protocolo SFTP (SSH File Transfer Protocol) o el servicio SCP (Secure Copy Protocol). SFTP es un estándar del paquete SSH-2. SCP es un parte estándar del paquete SSH-1. Ambos protocolos pueden ser empleados en ambas versiones de SSH. WinSCP puede ser usado tanto con servidores SSH-1 como SSH-2.

Interfaz del programa

WinSCP permite escoger entre dos tipos de interfaces, con muchas opciones de configuración para cada una de las, o que ofrece un alto grado de personalización.

Se le propondrá la selección del tipo de interfaz durante la instalación. Aunque es posible modificar las opciones posteriormente.

Si ahora usa WinSCP por primera vez, es posible que quiera emplear la interfaz tipo Explorador de Windows, ya que es la más cercana al modo en que Windows trabaja. Si

está familiarizado con gestores de archivos más avanzados del tipo Norton Commander, tales como Total Commander, FAR, o Altap Salamander, escoja la interfaz estilo Norton Commander (paneles duales) ya que está pensada para ofrecer un mayor nivel de funcionalidad y rápido acceso a comandos de teclado. Es posible usarla incluso sin manejar el ratón. Si se acostumbra a esta opción, descubrirá como realizar operaciones de modo mucho más rápido. [19]

2.10 Brackets

Brackets es un nuevo proyecto de código abierto fundado por Adobe para impulsar el desarrollo de herramientas en la web. Basado en el principio de que los mejores desarrolladores para crear herramientas para desarrolladores web son de hecho desarrolladores web, Brackets está construido en JavaScript, HTML y CSS. Con licencia de MIT y alojado en github, Brackets es un proyecto que desafía el status que no solo en cómo se construye, sino también con sus ideas innovadoras.

Hay muchos editores de texto por ahí, pero no muchos se preocupan exclusivamente por los desarrolladores web. Los soportes juegan favoritos y están optimizados para la web moderna. Es para los diseñadores que están tan cómodos escribiendo HTML y CSS como lo son en Illustrator y Photoshop. También es para los desarrolladores de JavaScript que crean aplicaciones complejas dirigidas no solo al navegador, sino a todas partes donde se ejecuta la web abierta.

Debido a que está integrado en JavaScript, HTML y CSS, si usa corchetes, tiene las habilidades necesarias para personalizarlo, ampliarlo y contribuir a él. Fácil de bifurcar y con una licencia flexible y abierta, realmente es su editor. De hecho, cuando abra Brackets por primera vez, verá el código fuente de Brackets. En el equipo, usamos Brackets para desarrollar Brackets en este extraño tipo de estilo de inicio-yin-y-yang.

Otro beneficio increíble de desarrollar con estándares web abiertos es que Brackets tiene el potencial de ejecutarse en casi todas partes. Si bien primero nos enfocamos en el escritorio, esperamos complementar esa versión con una versión de Brackets que pueda ejecutarse exclusivamente en su navegador. También tenemos algunas ideas sobre cómo Brackets podría convertir su tableta en su entorno de desarrollo. Los soportes también podrían estar integrados en sus aplicaciones web existentes. Con un poco de ayuda, todas estas versiones podrían desarrollarse en paralelo. Las

posibilidades están abiertas.

Además de cómo se está construyendo, Brackets tiene algunas ideas que lo hacen especial:

El equipo de Brackets no está abatido con paneles flotantes, barras de herramientas desordenadas y otras distracciones. Queremos que sea sencillo pero productivo. Creemos firmemente que solo debe ser usted y su código, y una de las formas en que Brackets mantiene su diseño mínimo es a través de una nueva idea llamada Edición rápida. Cada proyecto web consta de varios archivos HTML, CSS y JavaScript interrelacionados que normalmente se distribuyen en la parte superior de su editor en forma de pestañas. En sus editores comunes, su trabajo es mantener un mapa mental de cómo se interrelacionan todos estos archivos. Si desea aplicar o modificar un estilo en un elemento HTML, debe ubicar la hoja de estilo correcta; desplazarse alrededor o búsqueda rápida de la clase; editar; salvar; luego vuelva a su archivo original. En los corchetes, simplemente abre una Edición rápida presionando Cmd / Ctrl + E. Los corchetes arrancan el documento actual y muestra solo las clases de CSS que se aplican a ese elemento en línea. Es algo que hay que ver para creer.

También podemos aplicar las mismas ideas al desarrollo de JavaScript. Aplastar la tecla de edición rápida en una llamada de función y los corchetes abren un editor en línea con el cuerpo de la función de JavaScript listo para editar. En lugar de administrar documentos, los desarrolladores pueden ingresar y salir de su código con facilidad. [20]

2.11 Educación vial

La educación vial consiste en la adquisición de conocimiento por parte del ciudadano sobre la vía pública y la seguridad vial, a fin de desarrollar habilidades sobre el comportamiento como peatón o conductor de un vehículo en las calles de una población. Conocer las normas de tránsito permite evitar accidentes y pérdida de vidas que son consecuencia de situaciones fatales.

Los gobiernos promueven la difusión de estas normas y exigen como condición para circular la licencia o permiso para conducir. El conocimiento que pretende transmitir la educación vial, se refiere a los tipos, partes y características de la vía pública, las normas de circulación, señales de tránsito, las causas y factores de riesgo de los accidentes de tránsito, sus consecuencias y los primeros auxilios. Entre otros aspectos,

se hace hincapié en los elementos indispensables para garantizar la propia protección y la de terceros, incluyendo los aspectos legales a tener en cuenta ante determinadas infracciones.

Uno de los principales elementos que debe conocer un conductor es el cinturón de seguridad, que ante un impacto evita la colisión del conductor dentro el habitáculo, restringiendo todo movimiento. También son exigibles los apoya cabezas, espejos retrovisores, matafuegos, balisas entre otros. Del mismo modo se exigen los cascos para los motociclistas y sus acompañantes. La educación vial es un tema de mucha sensibilidad social, con respecto a la conducta vial a desarrollar para fomentar la convivencia, el respeto, la tolerancia, la responsabilidad y solidaridad que beneficien las relaciones humanas en la comunidad.

Como todo proceso educativo, la educación vial se inicia en el seno familiar, se imparte de padres a hijos, en las escuelas y todo centro docente. También las entidades públicas apoyan desde los ministerios públicos, ayuntamientos y dirección general de tránsito con campañas de promoción y prevención. Y las entidades privadas aportan recursos económicos para sustentar talleres, conferencias, libros y campañas publicitarias, entendiendo que la mejor prevención es una buena educación. [23]

2.12 jQuery

Para simplificar, podríamos decir que jQuery es un framework Javascript, pero quizás muchos de los lectores se preguntarán qué es un framework. Pues es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, framework son unas librerías de código que contienen procesos o rutinas ya listos para usar.

Los programadores utilizan los frameworks para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar. Nota: si no sabes lo que es Javascript seguramente no te interesará este artículo, pero puedes aprenderlo también en DesarrolloWeb.com: Qué es Javascript

Por ejemplo, en el caso que nos ocupa, jQuery es un framework para el lenguaje

Javascript, luego será un producto que nos simplificará la vida para programar en este lenguaje. Como probablemente sabremos, cuando un desarrollador tiene que utilizar Javascript, generalmente tiene que preocuparse por hacer scripts compatibles con varios navegadores y para ello tiene que incorporar mucho código que lo único que hace es detectar el browser del usuario, para hacer una u otra cosa dependiendo de si es Internet Explorer, Firefox, Opera, etc. jQuery es donde más nos puede ayudar, puesto que implementa una serie de clases (de programación orientada a objetos) que nos permiten programar sin preocuparnos del navegador con el que nos está visitando el usuario, ya que funcionan de exacta forma en todas las plataformas más habituales.

Así pues, este framework Javascript, nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con jQuery obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Cuando programemos Javascript con jQuery tendremos a nuestra disposición una interfaz para programación que nos permitirá hacer cosas con el navegador que estemos seguros que funcionarán para todos nuestros visitantes. Simplemente debemos conocer las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la consecución de nuestros objetivos.

Además, todas estas ventajas que sin duda son muy de agradecer, con jQuery las obtenemos de manera gratuita, ya que el framework tiene licencia para uso en cualquier tipo de plataforma, personal o comercial. Para ello simplemente tendremos que incluir en nuestras páginas un script Javascript que contiene el código de jQuery, que podemos descargar de la propia página web del producto y comenzar a utilizar el framework.

El archivo del framework ocupa unos 56 KB, lo que es bastante razonable y no retrasará mucho la carga de nuestra página (si nuestro servidor envía los datos comprimidos, lo que es bastante normal, el peso de jQuery será de unos 19 KB). Además, nuestro servidor lo enviará al cliente la primera vez que visite una página del sitio. En siguientes páginas el cliente ya tendrá el archivo del framework, por lo que no necesitará transferirlo y lo tomará de la caché. Con lo que la carga de la página sólo se verá afectada por el peso de este framework una vez por usuario. Las ventajas a la hora de

desarrollo de las aplicaciones, así como las puertas que nos abre jQuery compensan extraordinariamente el peso del paquete.

Ventajas de jQuery con respecto a otras alternativas

Es importante comentar que jQuery no es el único framework que existe en el mercado. Existen varias soluciones similares que también funcionan muy bien, que básicamente nos sirven para hacer lo mismo. Como es normal, cada uno de los frameworks tiene sus ventajas e inconvenientes, pero jQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones.

Además, es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otra cosa muy interesante es la dilatada comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc. [24]

Javascript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Muchos confunden el Javascript con el Java pero ambos lenguajes son diferentes y tienen sus características singulares. Javascript tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.

Java por su parte tiene como principal característica ser un lenguaje independiente de la plataforma. Se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado: Linux, Windows, Apple, etc. Debido a sus características también es muy utilizado para internet.

Como síntesis se puede decir que Javascript es un lenguaje interpretado, basado en prototipos, mientras que Java es un lenguaje más orientado a objetos.

2.13 ¿Qué es Javascript?

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes,

orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript.

¿Cómo nace Javascript?

Javascript nació con la necesidad de permitir a los autores de sitio web crear páginas que permitan intercambiar con los usuarios, ya que se necesitaba crear webs de mayor complejidad. El HTML solo permitía crear páginas estáticas donde se podía mostrar textos con estilos, pero se necesitaba interactuar con los usuarios.

En los años de 1990, Netscape creó Livescript; las primeras versiones de este lenguaje fueron principalmente dedicadas a pequeños grupos de diseñadores Web que no necesitaban utilizar un compilador, o sin ninguna experiencia en la programación orientada a objetos.

A medida que estuvieron disponibles nuevas versiones de este lenguaje incluían nuevos componentes que dan gran potencial al lenguaje, pero lamentablemente esta versión solo funcionaba en la última versión del Navigator en aquel momento.

En diciembre de 1995, Netscape y Sun Microsystems (el creador del lenguaje Java) luego de unirse objetivo de desarrollar el proyecto en conjunto, reintroducen este lenguaje con el nombre de Javascript. En respuesta a la popularidad de Javascript, Microsoft lanzó su propio lenguaje de programación a base de script, VBScript (una pequeña versión de Visual Basic).

En el año de 1996 Microsoft se interesa por competir con Javascript por lo que lanza su lenguaje llamado Jscript, introducido en los navegadores de Internet Explorer. A pesar de las diferentes críticas que se le hacen al lenguaje Javascript, este es uno de los

lenguajes de programación más populares para la web. Desde que los navegadores incluyen el Javascript, no necesitamos el Java Runtime Environment (JRE), para que se ejecute.

El Javascript es una tecnología que ha sobrevivido por más de 10 años, es fundamentales en la web, junto con la estandarización de la “European Computer Manufacturers Association” (ECMA) (adoptada luego por la ISO) y W3C DOM, Javascript es considerado por muchos desarrolladores web como la fundación para la próxima generación de aplicaciones web dinámicas del lado del cliente.

La estandarización de Javascript comenzó en conjunto con ECMA en Noviembre de 1996. Es adoptado este estándar en Junio de 1997 y luego también por la “Internacional Organization for Standardization” (ISO). El DOM por sus siglas en inglés “Modelo de Objetos del Documento” fue diseñado para evitar incompatibilidades.

¿Dónde puedo ver funcionando Javascript?

Entre los diferentes servicios que se encuentran realizados con Javascript en Internet se encuentran:

- Correo
- Chat
- Buscadores de Información

También podemos encontrar o crear códigos para insertarlos en las páginas como:

- Reloj
- Contadores de visitas
- Fechas
- Calculadoras
- Validadores de formularios
- Detectores de navegadores e idiomas

¿Cómo identificar código Javascript?

El código javascript podemos encontrarlo dentro de las etiquetas `<body></body>` de nuestras páginas web. Por lo general se insertan entre: `<script></script>`. También pueden estar ubicados en ficheros externos usando:

```
<script type="text/javascript" src="micodigo.js"></script>
```

Algunas características del lenguaje son:

Su sintaxis es similar a la usada en Java y C, al ser un lenguaje del lado del cliente este es interpretado por el navegador, no se necesita tener instalado ningún Framework.

- Variables: `var = "Hola", n=103`
- Condiciones: `if(i<10){ ... }`
- Ciclos: `for(i; i<10; i++){ ... }`
- Arreglos: `var miArreglo = new Array("12", "77", "5")`
- Funciones: Propias del lenguaje y predefinidas por los usuarios
- Comentarios para una sola línea: `// Comentarios`
- Comentarios para varias líneas:

```
/*  
Comentarios  
*/
```
- Permite la programación orientada a objetos: `document.write("Hola");`
- Las variables pueden ser definidas como: `string`, `integer`, `flota`, `boolean` simplemente utilizando `"var"`. Podemos usar `"+"` para concatenar cadenas y variables.

¿Es compatible con navegadores?

Javascript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros.

Con el surgimiento de lenguajes como PHP del lado del servidor y Javascript del lado del cliente, surgió Ajax en acrónimo de (Asynchronous Javascript And XML). El mismo es una técnica para crear aplicaciones web interactivas. Este lenguaje combina varias tecnologías:

- HTML y Hojas de Estilos CSS para generar estilos.
- Implementaciones ECMAScript, uno de ellos es el lenguaje Javascript.
- XMLHttpRequest es una de las funciones más importantes que incluye, que permite intercambiar datos asincrónicamente con el servidor web, puede ser mediante PHP, ASP, entre otros.[25]

DISEÑO METODOLÓGICO

3.1 Metodología de la investigación

El uso de la metodología quedo a cargo del ING. Domingo Marco Antonio Hernández Jiménez quien es el gerente general encargado de la supervisión del proyecto y fue el que decidió el uso de la metodología "Daily Scrum" por los motivos que se mencionan a continuación:

En Scrum , cada día de un sprint, el equipo organiza una reunión diaria llamada "scrum diario". Las reuniones se realizan normalmente en el mismo lugar y a la misma hora todos los días. Idealmente, una reunión diaria de scrum se lleva a cabo en el Por la mañana, ya que ayuda a establecer el contexto para el trabajo del día siguiente. Estas reuniones de scrum se limitan estrictamente a 15 minutos. Esto mantiene el debate enérgico pero relevante.

Scrum otorga un estatus especial a aquellos que están comprometidos, y muchos equipos aplican una regla en la que solo aquellos que están comprometidos pueden hablar durante la reunión diaria de scrum.

El equipo y Scrum Master se consideran comprometidos por casi todos los miembros de la comunidad Scrum. Hay algún desacuerdo sobre el propietario del producto. Mi opinión es que el propietario de un producto debe considerarse un participante dedicado del proyecto. (Y debe comportarse como uno, también.)

Todos los miembros del equipo deben asistir a las reuniones de scrum. Dado que tanto Scrum Master como el propietario del producto son miembros comprometidos del equipo, se espera que asistan y participen. Cualquier otra persona (por ejemplo, un vicepresidente departamental, un vendedor o un desarrollador de otro proyecto) puede asistir, pero solo está para escuchar. Esto hace que las reuniones de scrum sean una excelente manera para que un equipo de Scrum disemine información; si está interesado en saber dónde están las cosas, asista a la reunión de ese día.

La reunión diaria de scrum no se utiliza como una resolución de problemas o reunión de resolución de problemas. Los problemas que se plantean se toman fuera de línea y generalmente son resueltos por el subgrupo relevante inmediatamente después de la reunión. Durante el scrum diario, cada miembro del equipo responde las siguientes tres preguntas:

1. ¿Qué hiciste ayer?
2. ¿Qué vas a hacer hoy?
3. ¿Hay algún impedimento en tu camino?

Al centrarse en lo que cada persona logró ayer y logrará hoy, el equipo obtiene una excelente comprensión de lo que se ha hecho y lo que queda por hacer. La reunión diaria de scrum no es una reunión de actualización de estado en la que un jefe recopila información sobre quién está retrasado. Más bien, es una reunión en la que los miembros del equipo se comprometen entre sí.

Si un programador se pone de pie y dice: "Hoy terminaré el módulo de almacenamiento de datos", todo el mundo sabe que en la reunión de mañana, dirá si terminó o no. Esto tiene el maravilloso efecto de ayudar a un equipo a darse cuenta de la importancia de estos compromisos, y de que sus compromisos son los *unos con los otros*, no con un cliente o vendedor lejano.

En los casos en que el ScrumMaster no puede eliminar estos impedimentos directamente (por ejemplo, generalmente los problemas más técnicos), sigue siendo responsable de asegurarse de que alguien del equipo resuelva el problema rápidamente.

La gran mayoría de los equipos realizan la reunión diaria de scrum haciendo que cada persona responda las tres preguntas en orden. Responde a las tres, luego a la siguiente persona, a la siguiente y así sucesivamente. Una alternativa interesante que a algunos equipos les resulta útil es hablar sobre un elemento de la cartera de productos antes de pasar al siguiente. De esta manera, un individuo puede dar una actualización en múltiples momentos diferentes durante la misma reunión. [21]

The image shows a Google Calendar event card for 'Daily Scrum' on December 3rd, 2018. The event is scheduled for 11pm and is a weekly recurring event. The location is Skype. The organizer is Edgar Aguiar. Below the event card is a detailed description page for the event, which includes the objective, a list of three questions to be discussed, and the event details like time, location, and organizers.

Daily Scrum
 Miralo en Google Calendar

Cuándo Cada semana de 10pm a 11pm días de la semana (UTC)
 Ubicación Skype
 Participantes Edgar Aguiar, coding.smart.solutions@gmail.com*

Agenda
 lun 3 de dic de 2018
 No hay eventos anteriores.
 11pm Daily Scrum
 No hay eventos posteriores.

This event has been changed.

Daily Scrum [more details »](#)

El objetivo de estas reuniones es resolver cualquier problema que se tenga relacionado con el proyecto. La mecánica de las reuniones será básicamente responder a tres preguntas;

1. ¿Qué hice el día d ayer?
2. ¿Tengo algún impedimento para seguir avanzando en mis actividades?
3. ¿Qué voy hacer el día de hoy?

When **Changed:** Weekly from 10pm to 11pm on weekdays Coordinated Universal Time

Where Skype ([map](#))

Calendar mario romero

Who

- coding.smart.solutions@gmail.com - organizer
- mario romero
- Edgar Aguiar

Going? All events in this series: [Yes](#) - [Maybe](#) - [No](#) [more options »](#)

Ilustración 3 Daily Scrum

3.2 Desarrollo

El desarrollo del proyecto comenzó con el análisis de los requerimientos iniciales del proyecto. Éste fue realizado por el Ing. Marco antes del comienzo del proyecto de estadía. Por lo que comenzamos casi de inmediato con el desarrollo.

Primeramente se generaron los elementos base del proyecto esto es, todo lo que las personas puedan necesitar para hacer denuncias ciudadanas para mantener el orden vial en México:

- una forma interactiva en la que se puedan subir las ofensas viales.
- una forma rápida de poder encontrar las ofensas viales que se han subido.
- poder agregar personas para poder amigos agregados.
- crear un chat con el cual se pueda interactuar entre los diferentes usuarios de la página.

Estos requerimientos sirvieron como base para comenzar con el diseño de la página.

Una vez teniendo esto se siguió con la elección de las herramientas que se utilizarían para el desarrollo de la página web.

3.3 Instrumentos

Se decidió que el software se iba encontrar en web y el lenguaje acordado fue PHP, javaScrip y jqery. Con la implantación en matel desing para el diseño de la interfaz.

Estas herramientas fueron elegidas para tener un diseño más uniforme y controlado y un mejor manejo del desarrollo de la página web.

Para el desarrollo de la base de datos para guardar todo el contenido de los usuarios y de la página se decidió utilizar mysql y para el gestor de la base de datos se utilizo fue workbench ya que ofrece una buena interfaz para la manipulación del contenido de la base de datos (la imágenes solo son representativas).

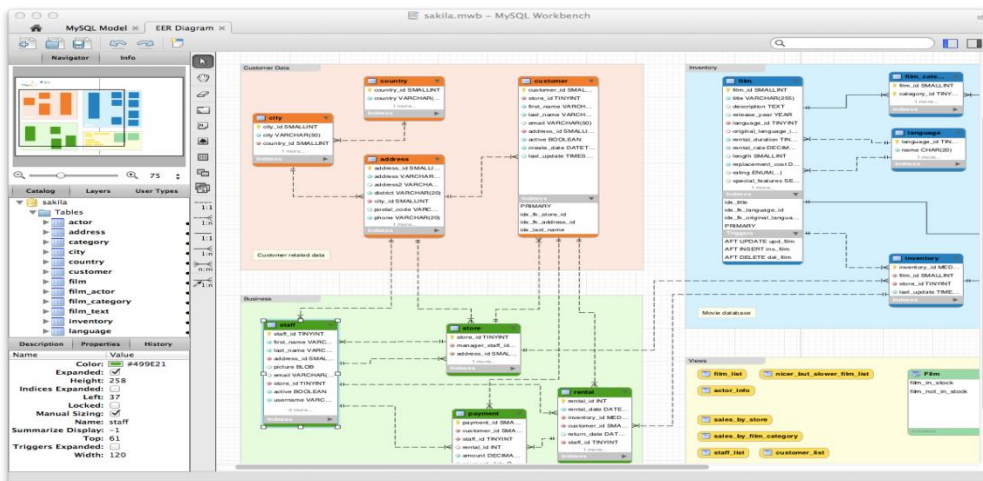


Ilustración 4 Workbench

Para el desarrollo del contenido de la página se utilizaron los apis de google map para el manejo de los mapas y así poder subir las denuncias en el lugar donde sucedieron.

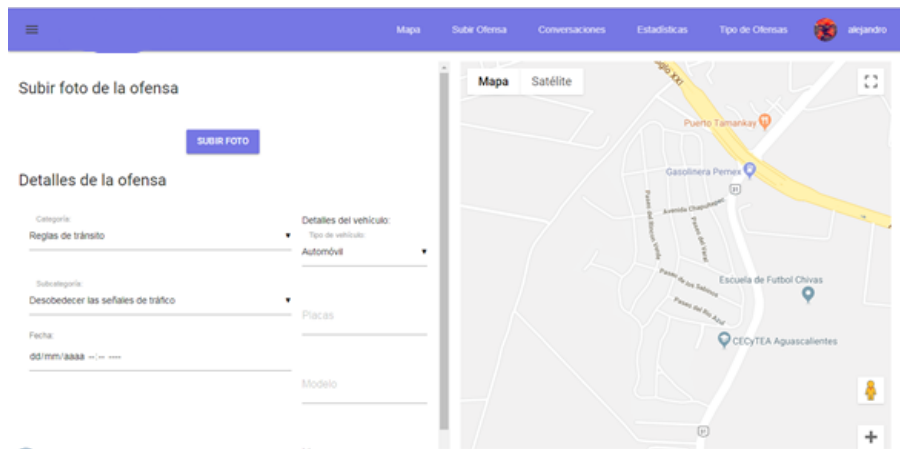


Ilustración 5 Ejemplo Pagina web

En cuanto a la codificación, se trabajó a través de FTP en un servidor local de desarrollo, utilizando el IDE braket. Se eligió este IDE más que nada por la practicidad, a la rapidez para comenzar a trabajar en él, además de ser de código libre.

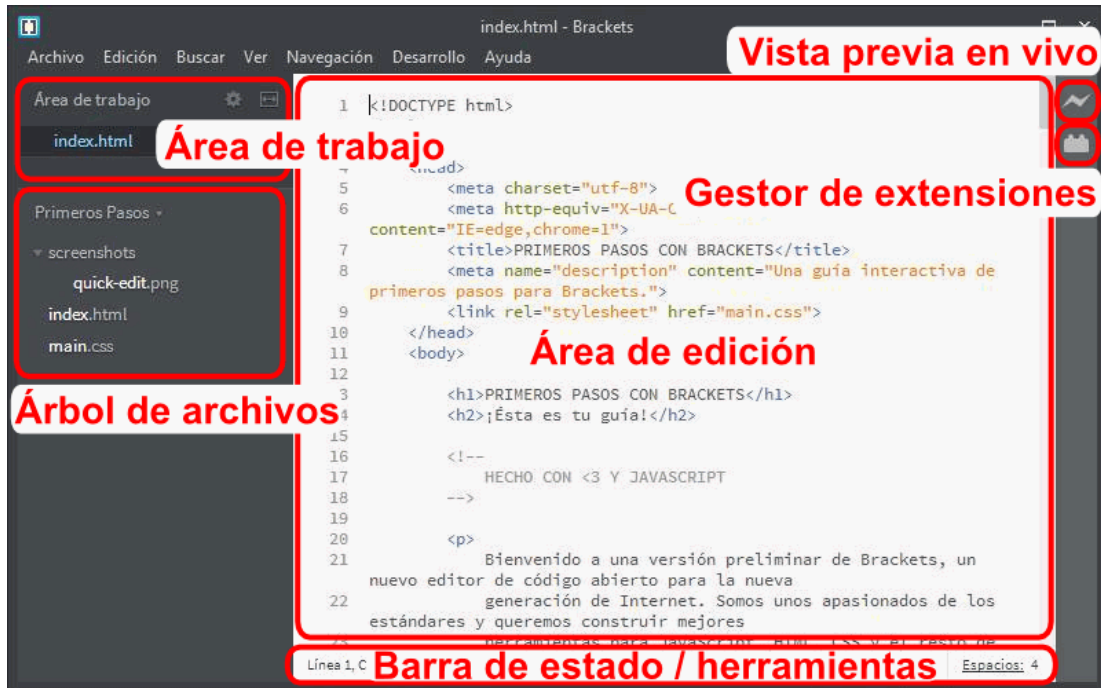


Ilustración 6 Braket

Desarrollo del chat

Para el desarrollo del chat se decidió utilizar firebase por su gran facilidad en el manejo de alta cantidad de datos en tiempo real.



Ilustración 7 Logo firebase

Para el desarrollo del chat se utilizó firebase para el almacenamiento en la base de datos para el funcionamiento del contenido se utilizó javaScrip y css para el estilo de diseño en el chat se pueden establecer conversaciones con los diferentes usuarios de la página y tiene la opción de poder mandar fotos en el contenido del chat.

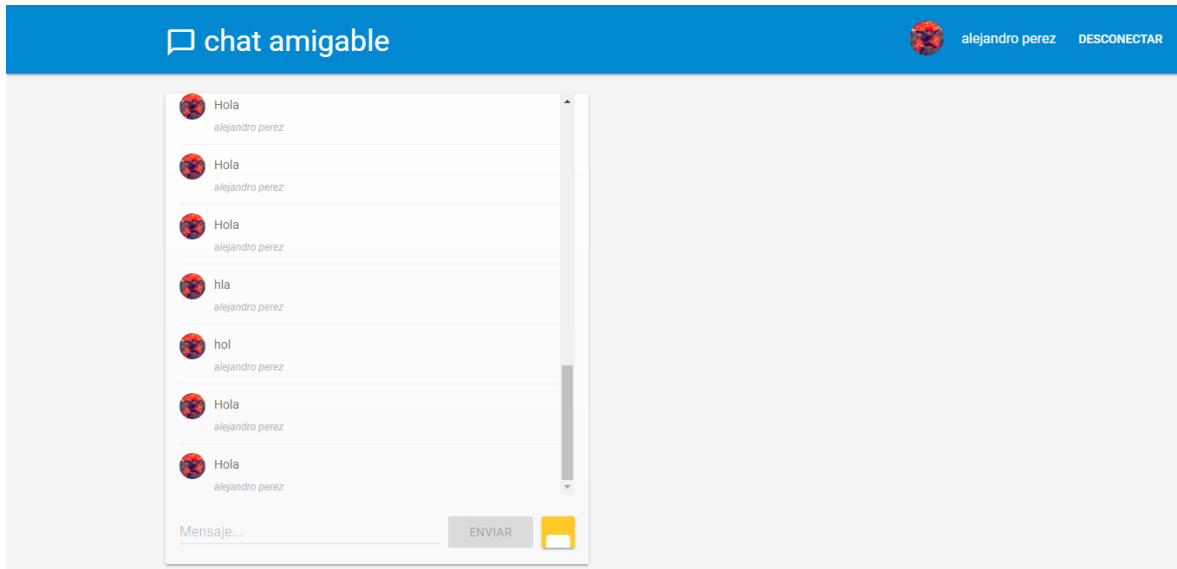


Ilustración 8 Ejemplo chat

CAPITULO IV

RESULTADOS DE LA INVESTIGACIÓN

4.1 Resultados

El resulta de la investigación y el desarrollo es una página web en la cual los mismos ciudadanos pueden hacer denuncias para que todos los usuarios de la página puedan ver los lugares y los automóviles que están cometiendo las faltas para que así entre la misma comunidad puedan buscar una solución para que las personas que cometen infracciones no se queden impunes y reciban un castigo por sus faltas a las reglas viales

Para subir una ofensa que sería la falta al reglamento vial se tiene que seleccionar la categoría de la ofensa que pueden ser reglas de tránsito, problemas con algún vehículo o conductor entre otras, luego una subcategoría para saber que se está haciendo como desobedecer en las señales de tráfico o pasarse un semáforo también se seleccionará la fecha el tipo de vehículo si es moto, automóvil, transporte público, peatón etc. Si es algún tipo de vehículo se llenaran las placas el modelo del carro, la marca y por ultimo un descripción de la infracción ya en el mapa se puede buscar el lugar donde se realizó y seleccionar haciendo un clic sobre el mapa para guardar la ubicación del lugar donde se cometió la infracción.

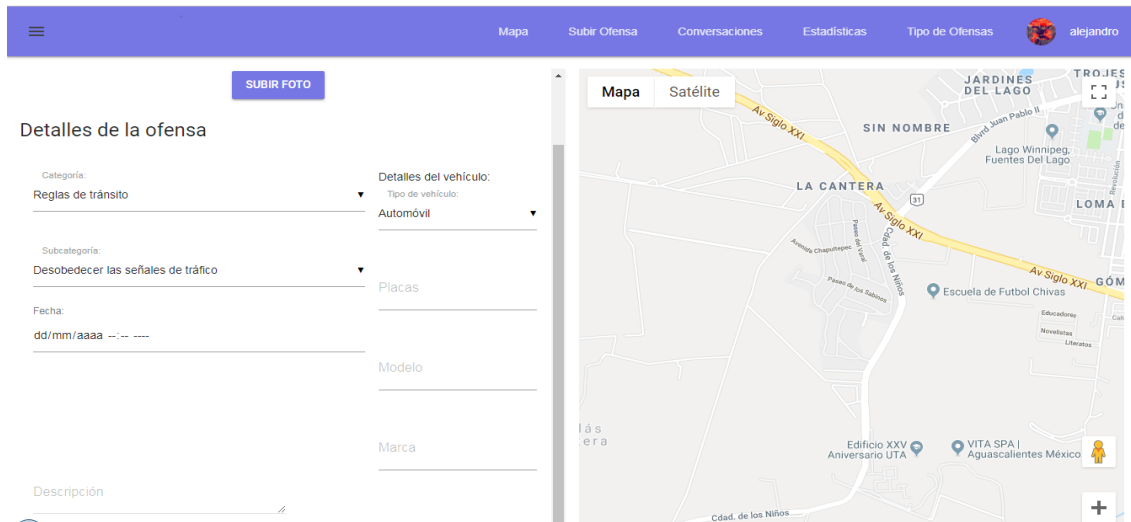


Ilustración 9 Ejemplo página web 2

Para buscar infracciones que se hayan cometido solo se tiene que ingresar la ubicación que quieres buscar y en el mapa se mostraran todas las infracciones que se hayan cometido en esa zona se puede filtrar por un radio de kilómetros que quieras buscar entre en la ubicación ingresada o ya de una forma más avanzada se puede buscar por

los diferentes filtro o categorías que se te dan al subir una ofensa como también por fechas y zonas.

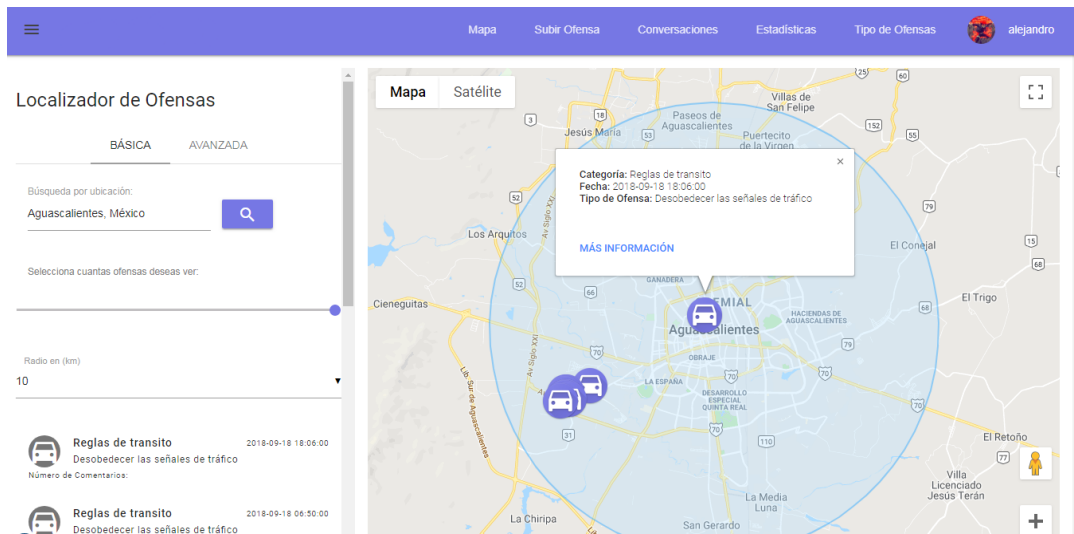


Ilustración 10 Ejemplo página web 3

Como resultado final de esta investigación se busca crear una comunidad en la cual los mismos ciudadanos entren en conciencia de las infracciones viales que están cometiendo

Página ha generado ayudas como en los lugares de discapacitados gente que de verdad no los necesita no se aprovechen de esos lugares y quede expuesta para que la próxima vez piense dos veces antes de ocupar un lugar que no le pertenece.

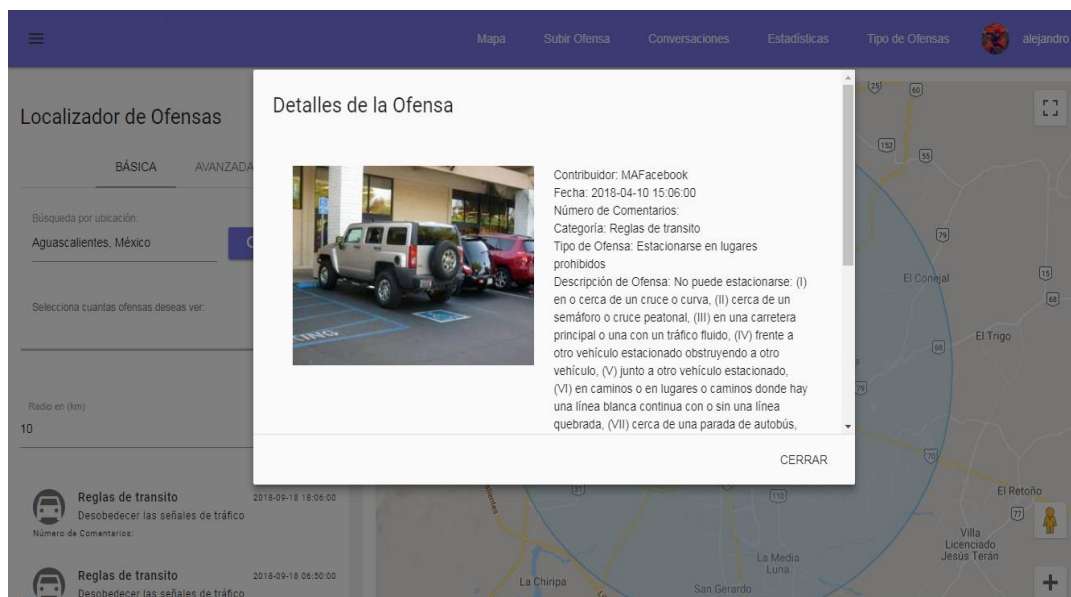


Ilustración 11 Ejemplo página web 4

CONCLUSIONES Y RECOMENDACIONES

En coding smart solutions se ha logrado tener un avance significativo, pero no total de lo que se quería lograr con el desarrollo de esta página web.

La página está actualmente alojada en un servidor de paga y cuenta con su propio dominio para ser buscada en internet y poder utilizar sus funcionalidades en cualquier dispositivo.

La página aún no está totalmente terminada, pero con las funcionalidades que tienen ya puede ser utilizada de manera óptima cumpliendo con el objetivo principal de ella solo con fallas en algunos detalles del diseño y de usabilidad menor que en un futuro pueden ser arreglados de forma óptima.

El chat una de las principales funciones de la página aún está en desarrollo, aunque ya está funcionando de manera óptima se busca mejorar de manera para que sea de mejor atractivo visual y tener más y mejores funcionalidades.

La página web no solo puede ser de ayuda para las personas si no también para las autoridades ya que en ella cualquier persona entra en el papel de oficial de tránsito buscando que se haga justicia en los infractores de las reglas viales. Las autoridades pueden entrar a la página y revisar lugares donde se comentan faltas viales y al tener evidencia poder hacer algo con respecto a eso.

Si esta página se hiciera oficial podría ayudar mucho a las autoridades de tránsito y que los infractores piensen dos veces antes de cometer una infracción ya que ahora no solo los oficiales de tránsito están pendientes si no también toda la sociedad.

Con el crecimiento de esta página se busca que las personas que estén más activas y publiquen más puedan recibir una premiación para que sea reconocido su esfuerzo y no solo eso para que las demás personas tengan un incentivo para unirse a esta causa de crear más conciencia al volante.

La información quedará a disposición de quien quiera buscarla. No se busca hacer lucro con esa información la información estará ahí para que las personas que quieran hacer algo al respecto tengan fuentes para informarse y pruebas para hacer alguna denuncia.

Esta página puede ser de mucha ayuda y se busca que llegue a muchas personas para así poder crear una comunidad más responsable al volante.

BIBLIOGRAFÍA

- [1] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. Thomas, D. (2001). Manifiesto 9for Software Agile Development [Internet], Disponible desde <http://agilemanifesto.org/> [Acceso junio 1, 2013].
- [2] Sommerville, I. Software engineering [9ª ed.]. Addison Wesley, Boston, 2010
- [3] Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. New Directions on Agile Methods: A Comparative Analysis. Proceedings 25th International Conference on Software Engineering. Portland, IEEE, 2003.
- [4] Pressman, R. S. Ingeniería del software: un enfoque práctico. McGraw-Hill, México, 2005.
- [5] IEEE Standards Association IEEE Standard 830-1998. IEEE recommended practice for software requirements specifications. IEEE Computer Society, Washington DC, 1998.
- [6] Hasnain, E. An Overview of Published Agile Studies: A Systematic Literature Review. Proceedings of the 2010 National Software Engineering Conference. New York, ACM, 2010.
- [7] Jiang, L., & Eberlein, A. Towards A Framework for Understanding the Relationships between Classical Software Engineering and Agile Methodologies. In Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral (APOS '08). New York, ACM, 2008.
- [8] Thorstein, M., Hannay, J., Pfahl, D., Benestad, H., & Langtangen, H. A literature review of agile practices and their effects in scientific software development. Proceedings of the 4th International Workshop on Software Engineering for Computational Science and Engineering (SECSE '11). New York, ACM, 2011
- [9] Silva, T., Selbach, S., Maurer, F., & Hellmann, T. User experience design and agile development: from theory to practice. Journal of Software Engineering and Applications, 5(10), 743-751, 2012.
- [10] Maurer, F. & Hellmann, T. (2013). People-Centered Software Development: An Overview of Agile Methodologies. En Lecture Notes in Computer Science, 7171, Berlín,

Springer, p.185-215.

[11] Vijayasarathy, L. & Turk, D. Agile software development: a survey of early adopters. *Journal of Information Technology Management* Volume, 19(2), 1-8, 2008.

[12] Hoda, R., Kruchten, P., Noble, J., & Marshal, S. *Agility in Context*. Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA '10). New York, ACM, 2010.

[13] anónimo, a. (2018). *Qué es SCRUM*. [online] *Proyectos Ágiles*. Available at: <https://proyectosagiles.org/que-es-scrum/> [Accessed 30 Nov. 2018].

[14] Firebase. (2018). *Firestore*. [online] Available at: <https://firebase.google.com> [Accessed 30 Nov. 2018].

[15] Pérez Capdevila., J. (2018). *Las Tecnologías Web para la Gestión del Conocimiento*. [online] *SociedadelaInformacion.com*. Available at: http://www.sociedadelaInformacion.com/9/las_tecnologias_web.htm [Accessed 1 Dec. 2018].

[16] A. Cobo, P. Gómez, D. Pérez, R. Rocha. "EL LENGUAJE PHP". En *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos, 2005. pp.99-124.

[17] Gilfillan, I. (2003). *La biblia de MySQL*. Madrid: Anaya Multimedia.

[18] Merino, M. (2018). *¿Qué es una API y para qué sirve?* [online] *TICbeat*. Available at: <http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/> [Accessed 1 Dec. 2018].

[19] Winscp.net. (2018). *Introducción: WinSCP*. [online] Available at: <https://winscp.net/eng/docs/lang:es> [Accessed 1 Dec. 2018].

[20] Blog.brackets.io. (2018). *Brackets Blog - The Free, Open Source Code Editor for the Web*. [online] Available at: <http://blog.brackets.io/2012/06/25/brackets-open-source-code-editor/> [Accessed 1 Dec. 2018].

[21] Cohn, M. (2018). *The Daily Scrum Meeting*. [online] *Mountain Goat Software*. Available at: <https://www.mountaingoatsoftware.com/agile/scrum/meetings/daily-scrum> [Accessed 1 Dec. 2018].

[22] Valdez, J. (2007). *Metodologías del Desarrollo de Software*. [online] *OK HOSTING | Hospedaje Web, Dominios, Desarrollo de Software, Marketing Online, SEO*. Available

at: <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/> [Accessed 15 Nov. 2018].

[23] Que Significado. (2018). *Significado de Educación vial - Qué es, Definición y Concepto*. [online] Available at: <https://quesignificado.com/educacion-vial/> [Accessed 1 Dec. 2018].

[24] Alvarez, M. (2018). *Introducción a jQuery*. [online] DesarrolloWeb.com. Available at: <https://desarrolloweb.com/articulos/introduccion-jquery.html> [Accessed 1 Dec. 2018].

[25] Pérez Valdés, D. (2007). *¿Qué es Javascript?*. [online] Maestros del Web. Available at: <http://www.maestrosdelweb.com/que-es-javascript/> [Accessed 1 Dec. 2018].

[26] Salud, S. (2018). *Informe sobre la situación de la seguridad vial, México 2016*. [online] gob.mx. Available at: <https://www.gob.mx/salud/acciones-y-programas/informe-sobre-la-situacion-de-la-seguridad-vial-mexico-2016?state=published> [Accessed 7 Dec. 2018].

[27] Excélsior. (2018). *En la Ciudad de México hay más de 4 millones de autos*. [online] Available at: <https://www.excelsior.com.mx/comunidad/2016/09/22/1118438> [Accessed 7 Dec. 2018].

[28] Seguros Bancomer. (2018). *Las infracciones de tránsito más comunes | BBVA Bancomer Seguros*. [online] Available at: <https://www.segurosbancomer.com.mx/las-infracciones-de-transito-mas-comunes/> [Accessed 11 Dec. 2018].

Glosario

PHP: Hypertext Preprocessor. Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

www: Sigla de la expresión inglesa *World Wide Web*, 'red informática mundial', sistema lógico de acceso y búsqueda de la información disponible en Internet, cuyas unidades informativas son las páginas web.

Scrum: es el nombre con el que se denomina a los marcos de desarrollo ágiles caracterizados por: Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.

API:(siglas de 'Application Programming Interface') es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

Vual: De los caminos, carreteras y vías públicas en general o relacionado con ellos.

Metodología: del griego *metà* (más allá), *odòs* (camino) y *logos* (estudio), hace referencia al conjunto de procedimientos basados en principios lógicos, utilizados para alcanzar una gama de objetivos que rigen en una investigación científica o en una exposición doctrinal