

BADWORK (control de acceso)

**2016**

Carlos Cañas Agenjo

IES Virgen de la Paz

01/01/2016



Índice

[1. Justificación del proyecto y objetivos 3](#_Toc447465731)

[2. Introducción 4](#_Toc447465732)

[3. Metodología y desarrollo del proyecto 5](#_Toc447465733)

[4. Resultado y discusión 6](#_Toc447465734)

[5. Conclusiones 7](#_Toc447465735)

[6. Bibliografia y referencias 8](#_Toc447465736)

[7. Anexos/Otros 9](#_Toc447465737)

# Justificación del proyecto y objetivos.

Los navegadores populares Chrome y Firefox nos dan la posibilidad de añadir mejoras y personalizarlos al uso cotidiano de cada usuario, ésta característica es lo que me ha llevado a la elección de este proyecto.

Estos programas que instalamos en nuestros navegadores, hacen de ellos una herramienta muy potente debido a que no sólo realizan su uso funcional, sino que añaden mejoras y pueden tener diferentes usos y características.

El proyecto que voy a realizar será un control de acceso, centrado en la actividad laboral de una empresa, cuya función es que el navegador bloquee las páginas no deseadas por el administrador.

Mi elección ha sido debida a que en los últimos años se ha visto un aumento de bajo rendimiento laboral debido a la libertad que nos da internet, lo que conlleva que los empleados hagan un mal uso de las nuevas tecnologías y en sus horas de actividad profesional no sólo accedan a contenido relacionado con el desarrollo de su profesión, sino también a contenido personal y de ocio.

Los objetivos que me planteo para dicho proyecto son los siguientes:

-Principal:

1. Bloquear las páginas no deseadas por el administrador.

-Secundarios:

1. Prevenir el mal uso de las nuevas tecnologías.
2. Aumentar el rendimiento de los empleados, por no tener acceso libre a internet.

# Introducción

El control de acceso es una herramienta que se puede habilitar por parte del administrador de forma rápida y sencilla, además de personalizarlo según las necesidades de cada usuario.

Su funcionalidad es filtrar el contenido web y la actividad de registro.

La habilitación y deshabilitación de dicho control de acceso normalmente está bloqueada por una contraseña que únicamente conoce el administrador.

Las principales características del control de acceso son:

* Bloqueo de páginas web: restringe el acceso a contenido inapropiado por medio del bloqueo de ciertos sitios web
* Restricción de acceso por categorías: aparecen en conjuntos, se muestra un listado donde se selecciona la categoría deseada a bloquear.
* Registro de actividad: permite observar un registro detallado de la actividad dentro del control de acceso.
* Selección rápida: se crean usuarios y según cada rol, se le permite el acceso a determinado contenido o no. Ejemplo: niño, adolescente, padre.
* Control de tiempo: limita el tiempo del ordenador conectado a internet.

El control de acceso es aplicable a diversos campos:

* Control en el hogar familiar, dónde los padres serían los administradores.
* Control escolar, en los centros educativos, bloqueando ciertos contenidos inapropiados e innecesarios durante las horas lectivas.
* Control laboral, en las empresas, para que así los empleados no hagan un mal uso de las nuevas tecnologías ni abusen del libre acceso por internet.

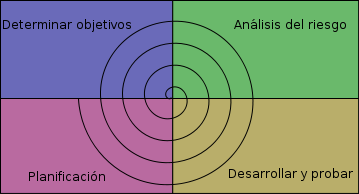
Mi proyecto de control de acceso se va a centrar en un control laboral, dentro de los ordenadores de una empresa, realizando un bloqueo de páginas web, para que los empleados puedan acceder sólo a los contenidos necesarios que se relacionen con su desarrollo laboral y así conseguir un aumento de su rendimiento. Se llamará “BADWORK” (mala palabra).

# Metodología y desarrollo del proyecto

METODOLOGÍA

La metodología que voy a seguir para desarrollar este proyecto será un modelo en espiral.

Cada bucle representa un conjunto de actividades, esto formará una caracola, que tiene dos dimensiones, la angular (indica el avance del proyecto) y la radial (indica el aumento de costes, porque por cada nueva interacción quiere decir que se pasa más tiempo desarrollando).

[](https://www.google.es/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwjig5HqlJbMAhVElxoKHaXhAH8QjRwIBw&url=https://es.wikipedia.org/wiki/Desarrollo_en_espiral&psig=AFQjCNHyma1VyqUC40jiTRmyewAZ2VANDQ&ust=1460999311743809)

Cada ciclo tendrá:

-Objetivos.

-Análisis de riesgos y alternativas.

-Desarrollo y prueba.

-“Planificación”, donde se probará finalmente y se verá si en necesario comenzar un nuevo ciclo.

El modelo en espiral de divide en regiones de tareas:

-Comunicación con el cliente.

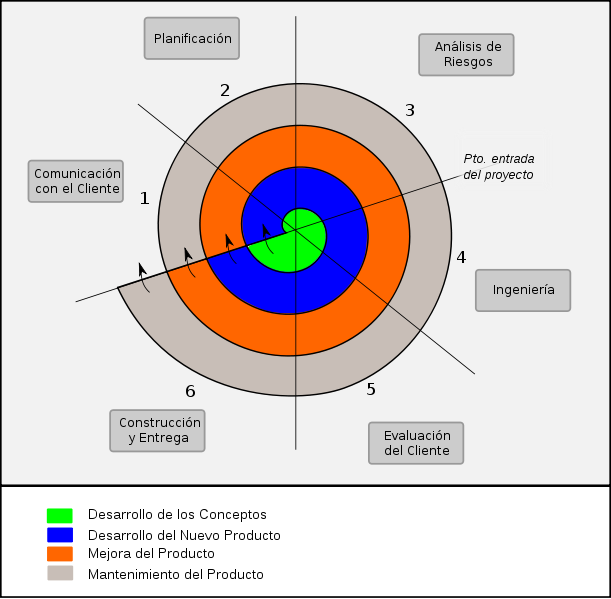
-Planificación.

-Análisis de riesgos.

-Ingeniería: tareas para construir representaciones de la aplicación.

-Construcción y acción: tareas para construir, probar, instalar y proporcionar soporté.

-Evaluación del cliente.



***DESARROLLO DEL PROYECTO***

**PRIMERA VERSIÓN (BADWORK 1.0)**

1. Comunicación con el cliente / Objetivo.

El cliente solicita que realice un control de acceso para su empresa, su intención es que los empleados no puedan acceder a las páginas web de redes sociales.

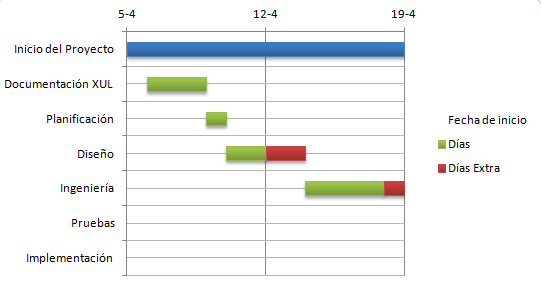
Objetivo: bloquear y desbloquear la página actual, y que esto quede guardado en la sesión.

1. Planificación.

En quince días le mostraré al cliente la primera versión del proyecto, y valoraremos si quiere continuar añadiendo mejoras.

Diagrama Gantt:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tareas | Fecha de inicio Programada | Fecha de inicio | Días | Días Extra | Fecha fin |
| Inicio del Proyecto | 05/04/2016 | 05/04/2016 | 14 | 0 | 19/04/2016 |
| Documentación XUL | 06/04/2016 | 06/04/2016 | 3 | 0 | 08/04/2016 |
| Planificación | 09/04/2016 | 09/04/2016 | 1 | 0 | 09/04/2016 |
| Diseño | 10/04/2016 | 10/04/2016 | 2 | 2 | 13/04/2016 |
| Ingeniería | 11/04/2016 | 14/04/2016 | 4 | 2 | 19/04/2016 |
| Pruebas | 15/04/2016 |  | 2 | 0 |  |
| Implementación | 17/04/2016 |  | 2 | 0 |  |



1. Análisis de riesgo

Comienzo el proyecto buscando información sobre los controles de acceso, el tipo de tecnología usada habitualmente, una de las dificultades que se me presentan es que la mayoría de información viene en lengua extrajera. Por lo que veo el idioma como un posible riesgo.

Otro de los riesgos que me planteo es encontrarme con versiones no actualizadas de la tecnología a utilizar para desarrollar el proyecto, como alternativa a esto me propongo intentar realizar el proyecto con la tecnología más óptima.

Retraso en los plazos de entrega por la necesidad de solventar diferentes riesgos.

1. Desarrollo y prueba

Para desarrollar este proyecto usaré las siguientes herramientas:

-Editor de texto: sublime text.

-IDE (integrated developement environment) “entorno de desarrollo interactivo”: XUL Explorer, facilita la creación de ficheros con extensión “.xul”.

-Navegador: Mozilla Firefox, con las siguientes extensiones: firebug, remote xul manager, developer assistant y colorzilla.

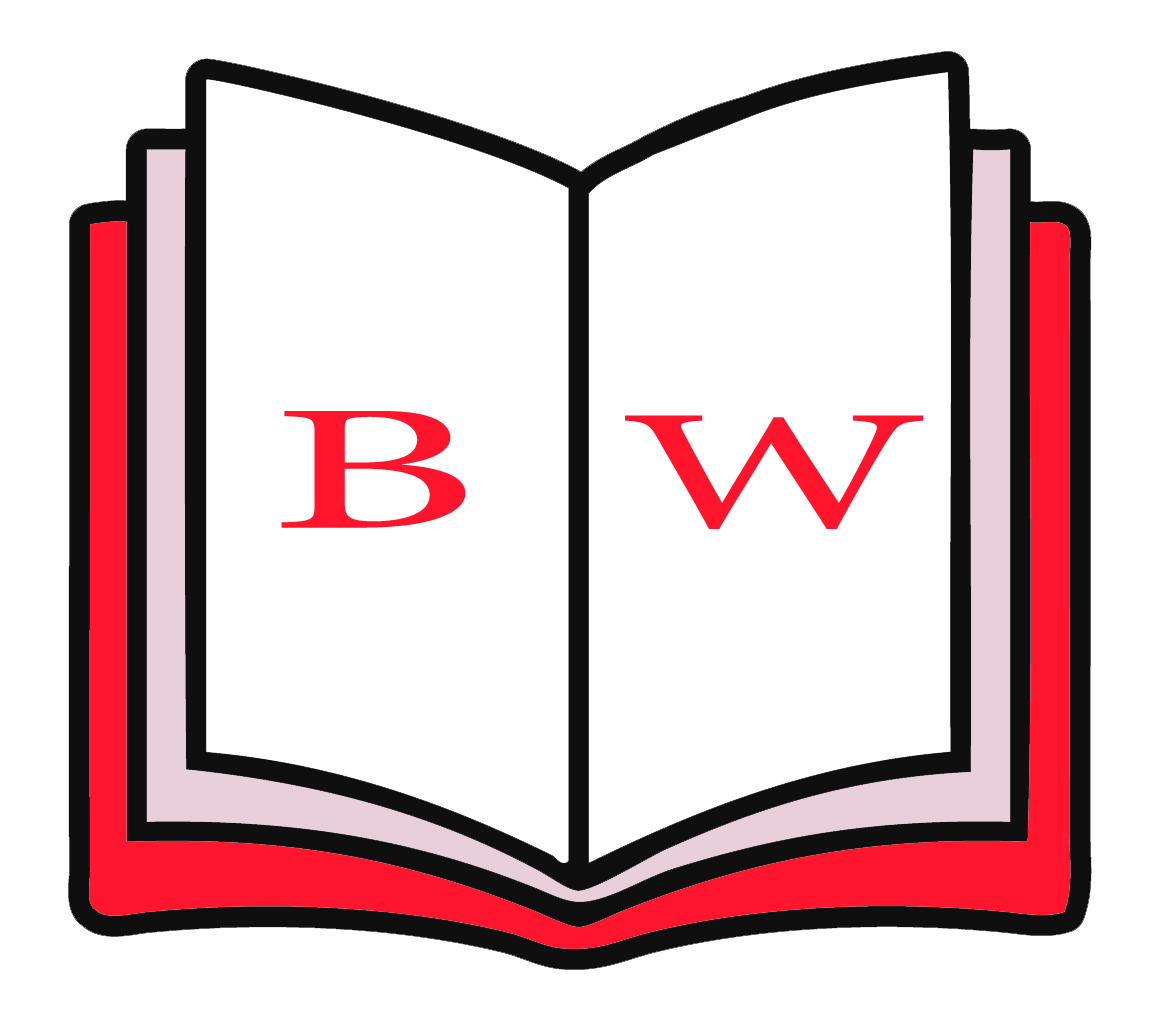
La estructura a seguir:

- Crear un archivo “.manifest” e “install.rdf” para poder generar el programa y probarlo.

- Comprimir todo el proyecto en formato “.zip” y luego cambiarlo a formato “.xpi”.

- Comprimir el contenido del proyecto excepto “.manifest” e “install.rdf” en formato”.zip” y cambiarlo a formato “.jar”.

Para empezar el desarrollo lo primero que hago es pensar un nombre para el programa, ya que el objetivo es evitar distracciones en el empleo creo que una buena posibilidad es llamarlo “BADWORK”, el símbolo elegido para representarlo es el siguiente:



Tras documentarme decido utilizar para este proyecto la tecnología “XUL”. Al empezar a programar con esta tecnología comienzo a encontrarme diversas dificultades:

- Cuando trabajo localmente con los archivos con extensión “.xul” hayo incompatibilidades con las versiones más actuales de Firefox.

- A la hora de instalar el API (Application Programming Interface) necesita la firma de verificación del grupo de desarrollo de Mozilla.

Por estos inconvenientes me decido finalmente realizar el proyecto en una tecnología que no esté tan desfasada e implementar el proyecto utilizando Add-on SDK (Kit de Desarrollo Software) que te permite desarrollar APIs de Firefox siguiendo las tecnologías Web estándar: JavaScript, Css y Html.

Este paquete de herramientas de desarrollo tiene numerosas ventajas respecto a la anterior tecnología:

* Posee una herramienta por línea de comando denominada **JPM**, que tiene diferentes funciones:
  + Inicializar un proyecto desde cero (creando una estructura base de directorios y archivos). **Jpm init**
  + Ejecutar el proyecto en local. **Jpm run**
  + Probar
  + Empaquetar proyecto para posteriormente instalarlo. **Jpm xpi**
* JPM está basada en Node.js y sustituye a la herramienta CFX.
* JPM se puede utilizar en versiones de Firefox 38 en adelante.
* Posee módulos ya definidos que te ayudan a la hora de crear el API e interactuar con el usuario.
* Puedes añadir módulos de terceros o incluso poner tus creaciones.

1. Evaluación y próxima planificación.

El cambio de tecnología ha sido un riesgo que hemos solventado pero ha conllevado un retraso en la entrega acordada según el tiempo estimado, fue comentado con el cliente en el momento oportuno, el cual no puso ningún inconveniente y decidió que continuásemos con el proyecto usando **ADD-ONS SDK.**

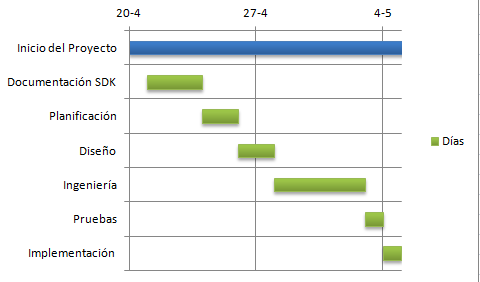
**PRIMERA VERSIÓN (BADWORK 1.1)**

1. Comunicación con el cliente / Objetivo.

Crear pequeña aplicación instalada en el navegador Firefox, con un botón que al pulsarlo haga que aparezca un panel con dos botones para bloquear y desbloquear la página actual.

1. Planificación.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tareas | Fecha de inicio Programada | Fecha de inicio | Días | Días Extra | Fecha fin |
| Inicio del Proyecto | 20/04/2016 | 20/04/2016 | 16 | 0 | 05/05/2016 |
| Documentación SDK | 21/04/2016 | 21/04/2016 | 3 | 0 | 23/04/2016 |
| Planificación | 24/04/2016 | 24/04/2016 | 2 | 0 | 25/04/2016 |
| Diseño | 26/04/2016 | 26/04/2016 | 2 | 0 | 27/04/2016 |
| Ingeniería | 28/04/2016 | 28/04/2016 | 5 | 0 | 02/05/2016 |
| Pruebas | 03/05/2016 | 03/05/2016 | 1 | 0 | 03/05/2016 |
| Implementación | 04/05/2016 | 04/05/2016 | 1 | 0 | 04/05/2016 |



1. Análisis de riesgo

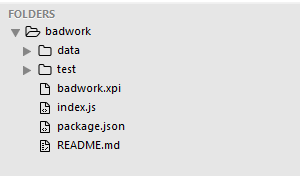
No conseguir almacenar en sesión las páginas bloqueadas.

No llegar a los plazos acordados en la planificación según las tareas.

1. Desarrollo y pruebas

Estructura:

Con el comando **JPM INIT** hemos creado una estructura base a la que le vamos a añadir diferentes cambios y explicar los diferentes directorios y archivos:



**Badwork:** carpeta raíz.

**Data:** carpeta donde se van almacenar las imágenes, js, html y css.

**Test:** carpeta que va a almacenar ficheros js para hacer pruebas.

**Badwork.xpi:** será el fichero compilado preparado para instalar.

**Index.js:** será el fichero que llevara la lógica de la API.

**Package.json:** archivo que describe características de la API, autor, colaborador, versión soportada, descripción, nombre, etc.

**README.md:** pequeña descripción del funcionamiento de la API.

Creación del icono-botón de la aplicación:

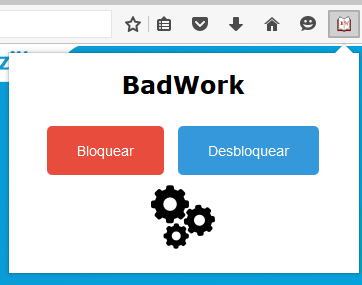
- Se almacena en la carpeta data tres imágenes con distinto tamaño:

* Icon-16.png

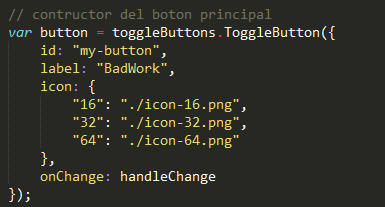


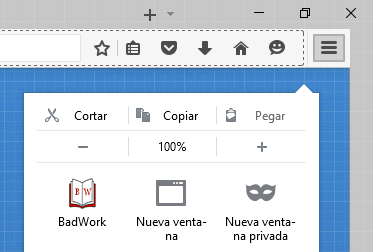
* Icon-32.png
* Icon-64.png

- Se crea un fichero panel.html y una hoja de estilos style.css en la carpeta data para dar el siguiente aspecto:



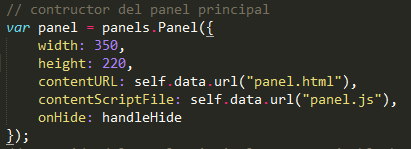
- Se crean los diferentes constructores que dependen de los módulos (botón, panel).

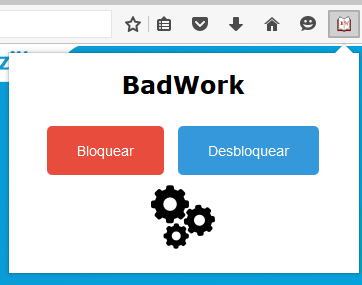


- Se construye el botón y se le asigna un id, un label, las imágenes que están en la carpeta data y una función que al pulsarlo haga que se muestre el panel.

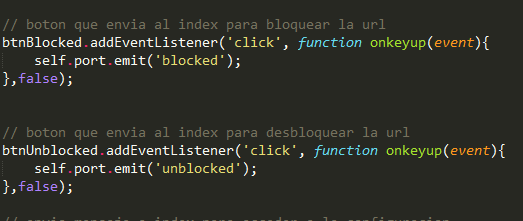
Screenshot_3.png

- Se construye el panel donde especificamos el ancho y alto, el contenido del panel (que será el archivo panel.html), también irá asociado un archivo javaScript y una función que hace que sólo se active el panel cuando se pincha el icono, sino quedaría oculto.





Con el archivo panel.js añadimos eventos a los botones que al pulsarlos van a mandar un mensaje que será recogido por el archivo Index.js. Esto se utiliza para recoger las acciones que se produzcan en el panel e interactuar con el usuario. Los mensajes pueden ser codificados a través de un JSON, string, array y objects. En este caso mandaremos un string.



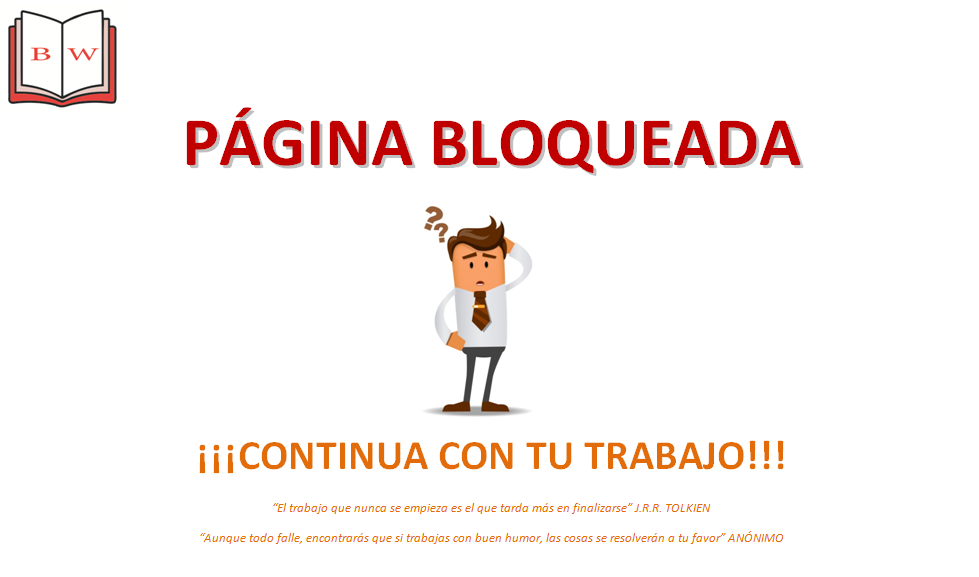
- Se construye un array donde se almacenarán todas las páginas que se bloqueen.

En index.js al recibir los mensajes ‘blocked’ se almacenará la página actual en el array y se bloqueará la página mostrando blocked.html.

Si por el contrario el mensaje es ‘unblocked’ se borrará la página del array y mostrará la página que fue bloqueada.

Diseño:

Para el diseño, me he centrado en algo llamativo y actual, para así sorprender al trabajador y que su reacción sea positiva, para animarle a que continúe con su actividad laboral y no se distraiga con el acceso libre que permite internet.



1. Evaluación y próxima planificación

Le propondré al cliente mejorar la versión actual, añadiendo la opción de configuración para poder visualizar en un listado de las páginas que ya ha bloqueado, y la creación -cambio de contraseña.

Cada vez que se bloquee aparecerán 2 citas de ánimo de manera aleatoria, previamente ya almacenadas.

Creación de un repositorio gitHub para mantener la aplicación. Investigar sobre incluir git en sublime.

# Resultado y discusión

# Conclusiones

# Bibliografía y referencias

<http://soporte.eset-la.com/kb2798/?locale=es_ES>

<https://es.wikipedia.org/wiki/Desarrollo_en_espiral>

<http://www.sites.upiicsa.ipn.mx/polilibros/portal/polilibros/P_externos/Administracion_informatica_de_las_organizaciones_Ramon_E_Enriquez_Gonzalez/AIO2_Mod_ESPIRAL.html>

<http://sofware1nathalygrijalva.blogspot.com.es/2012/10/modelo-espiral.html>

<http://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL/Tutorial>

<https://developer.mozilla.org/en-US/Add-ons/SDK>

<https://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software>

[https://developer.mozilla.org/en-US/Add-ons/SDK/Tools/jpm#Installation](https://developer.mozilla.org/en-US/Add-ons/SDK/Tools/jpm%23Installation)

<https://nodejs.org/en/>

# Anexos/Otros