**GUIA PARA IMPLEMENTACIÓN DE INTERFACES DE USUARIO**

**Elaborado por:** Equipo de proyecto Dashboard

**Dirigido a:** Estudiantes y desarrolladores del grupo de Investigación en Gestión de Proyectos

Sumario

[Introducción 1](#__RefHeading___Toc511_3673389285)

[Redmine 1](#__RefHeading___Toc513_3673389285)

[*Recursos para conocer sobre Redmine:* 2](#__RefHeading___Toc515_3673389285)

[React Native 3](#__RefHeading___Toc517_3673389285)

[Algunas ideas básicas de cómo usar la Api de Redmine 6](#__RefHeading___Toc519_3673389285)

[Aspectos generales: 6](#__RefHeading___Toc521_3673389285)

[Guías para HTML 9](#__RefHeading___Toc523_3673389285)

[Guías para CSS 10](#__RefHeading___Toc525_3673389285)

[Partes principales del CSS 10](#__RefHeading___Toc527_3673389285)

# Introducción

Las empresas necesitan gestionar sus proyectos y procesos de trabajo de forma rápida y eficiente, por lo que el uso de herramientas avanzadas en la gestión de proyectos supone una gran mejora en aspectos como la agilidad, el trabajo colaborativo y la productividad. Para ello se ha estado trabajando en el desarrollo de una plataforma para la Inteligencia de Negocios y la Gestión de Proyectos, cuya base principal es la plataforma web Redmine.

## Redmine

Es una herramienta multiplataforma liberada bajo los términos de la licencia GNU v2 (GPL) para la gestión y planificación de proyectos. Se trata de una solución altamente configurable que se adapta a cualquier tipo de empresa o proyecto. Esta herramienta incluye diferentes funcionalidades: seguimiento de incidentes, errores, calendario de actividades, diagramas de Gantt para representar visualmente la línea del tiempo de los proyectos, wiki, control basado en roles, integración con correo electrónico y un largo etcétera de características que resultan muy útiles. Además, Redmine puede instalarse en sólo unos pocos minutos. La última versión liberada es la 4.2.3

Características esenciales de Redmine son:

* Es una plataforma web, por lo que puede accederse desde cualquier dispositivo independientemente del sistema operativo que se utilice.
* Está orientado a la coordinación de tareas y a la gestión de flujos de trabajo.
* Facilita y fomenta el trabajo cooperativo y la comunicación de los participantes.
* Es personalizable con el uso de complementos y temas que permiten configurar su apariencia e incrementar el número de funciones disponibles.

Esta aplicación es multiplataforma (Unix, GNU/Linux, Mac, Mac Server y Microsoft Windows), además soporta diversas bases de datos (MySQL, PostgreSQL, Microsoft SQL Server, SQLite) por lo que se adapta a diversos entornos. Está desarrollada en diferentes lenguajes de programación y usa diferentes elementos asociados al desarrollo de aplicaciones web.

* Ruby 62.2%
* HTML 33.2%
* JavaScript 2.8%
* CSS 1.5%
* Other 0.3%

Para la convocatoria de investigación y desarrollo que se lanza se resaltan dos características fundamentales de la plataforma, que enfatizan el nivel de extensibilidad, adaptabilidad e integración de la misma:

* La capacidad de la misma de extender sus funcionalidades a través de complementos (plugins & themes)
* La capacidad de integrarse y permitir la integración a través de Api Rest y Web Hooks.

Estas dos características del Redmine son las que se consideran como base para el desarrollo que se está necesitando para perfeccionar la plataforma para la Inteligencia de Negocios y la Gestión de Proyectos, al lograr extender sus funcionalidades (análisis de datos, inteligencia de negocio, entre otras) e integrar nuevas tecnologías sobre todo las basadas en tecnologías móviles.

### *Recursos para conocer sobre Redmine:*

Su página oficial es: [https://www.redmine.org](https://www.redmine.org/)

Videos o minicursos: <https://ftp.prod.uci.cu/pub/Videos/Redmine>

Se pueden encontrar en la actualidad diversas formas para su despliegue, ya sean en entornos propios o en la nube. A continuación, se brindan referencias a sitios donde se pueden buscar instalaciones de Redmine:

* <https://bitnami.com/stack/redmine>

Se pueden encontrar una variedad de formas de despliegue de la plataforma:



* Docker: <https://hub.docker.com/_/redmine>
* Código fuente: [https://www.redmine.org](https://www.redmine.org/)
* Descripción de la Api Rest de Redmine:
	+ <https://www.redmine.org/projects/redmine/wiki/Guide#Developer-guide>
	+ <https://www.redmine.org/projects/redmine/wiki/Rest_api>
* Para conocer como desarrollar e instalar complementos para Redmine:
	+ <https://www.redmine.org/projects/redmine/wiki/Plugins>
	+ <https://www.redmine.org/projects/redmine/wiki/Themes>
	+ <https://www.redmine.org/projects/redmine/wiki/HowTo_create_a_custom_Redmine_theme>

## React Native

React Native es un framework JavaScript para crear aplicaciones reales nativas para iOS y Android, basado en la librearía de JavaScript React para la creación de componentes visuales, cambiando el propósito de los mismos para, en lugar de ser ejecutados en navegador, correr directamente sobre las plataformas móviles nativas, en este caso iOS y Andorid. Es decir, en lugar de desarrollar una aplicación web híbrida o en HTML5, lo que obtienes al final como resultado es una aplicación real nativa, indistinguible de la que podrías desarrollar con tu código en Java.

En la dirección URL que se muestra abajo podrán encontrar un grupo de videos que les ayudarán a entender y desarrollar aplicaciones con este framework.

* <https://ftp.prod.uci.cu/pub/Descargas/11_Noviembre/ReactNative>
* <https://ftp.prod.uci.cu/pub/Videos/Cursos/ReactJS>

Con esta idea de construcción de aplicaciones React Native nos proporciona las siguientes funcionalidades:

* **Compatibilidad Cross-Platform**: ya que la mayoría de las APIs de React Native lo son de por sí, lo cual ayuda a los propios desarrolladores a crear aplicaciones que puede ser ejecutados tanto en iOS como Android simultáneamente con el mismo código base.
* **Funcionalidad nativa**: las aplicaciones creadas mediante React Native funcionan de la misma manera que una aplicación nativa real creada para cada uno de los sistemas usando su lenguaje nativo propio. La unión de React Native junto con JavaScript permite la ejecución de aplicaciones más complejas de manera suave, mejorando incluso el rendimiento de las apps nativas y sin el uso de un WebView.
* **Actualizaciones instantáneas (para desarrollo y/o test)**: con la extensión de JavaScript, los desarrolladores tienen la flexibilidad de subir los cambios contenidos en la actualización directamente al dispositivo del usuario sin tener que pasar por las tiendas de aplicaciones propias de cada sistema y sus tediosos ciclos de procesos obligatorios previos. Hay que aclarar que este uno es exclusivo de versiones de desarrollo o para test, es ilegal, y puede llegar a conllevar castigos que llegan hasta la retirada definitiva de la aplicación si se realizan cambios directos sobre código con aplicaciones ya publicados y en producción. La tienda de Apple lleva un control muy exhaustivo sobre este tipo de prácticas.
* **Sencilla curva de aprendizaje**: React Native es extremadamente fácil de leer y sencillo de aprender ya que se basa en los conceptos fundamentales del lenguaje JavaScript, siendo especialmente intuitivo tanto para los ya expertos en dicho lenguaje o incluso para las personas sin experiencia en él, ya que nos provee de un rango muy amplio de componentes, incluyendo ejemplo como los maps y filters típicos que se han usado siempre.
* **Experiencia positiva para el desarrollador**: si bien la curva de aprendizaje hemos dicho que es sencilla, también el propio lenguaje nos motiva y ayuda a la hora de la evolución según aumentamos nuestro conocimiento y dominio del mismo. Nos ofrece varias características importantes como, por ejemplo, el Hot reloading que nos refresca la app en el momento en que guardamos cambios, y nos ofrece una gran ventaja para el desarrollo y testing de nuevas versiones, como hemos comentado arriba. O el uso del flexbox layout engine gracias al cual nos permite abstraernos de muchos de los tediosos detalles de la generación de cada uno de los layouts correspondientes a iOS y Android. Así como el uso del debugger de las herramientas de desarrollados del navegador Google Chrome, facilitando de sobre manera la tarea de depuración de código.

Estas son cinco de las características principales que nos brinda React Native por defecto si lo elegimos como framework para nuestro desarrollo. A priori suenan muy bien ¿verdad? Más adelante, a lo largo de una serie de artículos, veremos que no es oro todo lo que reluce, y que todas estas características no se cumplen siempre, o no al cien por cien, en todos los casos.

Para el desarrollo de las diferentes propuestas es necesario:

* Documentar con claridad los requisitos funcionales y no funcionales de la propuesta. Estos dos elementos son esenciales para definir el alcance de la propuesta.
* Diseño de cada vista (Se debe diseñar según el servicio las vistas que correspondan y además empaquetar los componentes)
* Desarrollar todas esas vistas previamente diseñadas.
* Realizar pruebas unitarias al código, de integración y seguridad, y se deberán realizar pruebas de funcionalidad para verificar el cumplimiento de los requisitos establecidos en la propuesta. A continuación el link de las pruebas <https://programacionymas.com/blog/tipos-de-testing-en-desarrollo-de-software>

Para el desarrollo se deberán estudiar:

* Las tecnologías y lenguajes a emplear.
* Servicios Api Rest de Redmine (cómo se consumen en React Native, que librerías se usan para eso)
* Estructura de los servicios REST (qué campos se gestionan, tipo de dato e ir buscando el diseño de los componentes en React, ejemplo la fecha con algún componente de calendario, entre otros)
* Si el desarrollo está relacionado con los complementos, se deberá tener claridad de cómo se estructuran internamente los diferentes tipos de complementos.

### Algunas ideas básicas de cómo usar la Api de Redmine

* Referencia: <https://www.redmine.org/projects/redmine/wiki/rest_api>

### Aspectos generales:

Para la Autenticación se emplean dos formas:

* Autenticación Básica: Usuario y Contraseña
* API Key – La herramienta asigna a cada usuario una clave de acceso de la API

Se emplean dos formatos de datos estructurados: XML o JSON.

Filtros generales que se usan en las peticiones y manejo de colecciones de datos:

* limit: Cantidad máxima de elementos solicitados
* offset: Elemento a partir de cual solicitar
* Existen otros filtros asociados a cada tipo de datos a solicitar, por ejemplo, para los users se pueden usar el filtro (status):

 status=1 para los usuarios activos

 status=2 para los usuarios registrados pero que aún no han sido confirmados

 status=3 para los usuarios bloqueados

En las solicitudes estos filtros deben estar contenidos, aunque no tengan valores, por ejemplo:

https://gp.prod.uci.cu/users.json?limit=15&offset=&status=&group\_id=&name=yordan

Esto devuelve a todos los que tengan yordan un su usuario, nombre o correo, sin importar el grupo al que pertenecen (group\_id), o tengan cualquier status.

Es preciso aclarar que, si no ponen los filtros, aunque sea sin valores los resultados no van a ser correctos.

Resultados:

* .xml - Formato XML, el tipo de contenido del resultado tiene que ser Content-Type: application/xml
* .json - Formato JSON, el tipo de contenido del resultado tiene que ser Content-Type: application/json

En cada solicitud de listado se devuelven estos elementos a parte del dato solicitado:

* Arreglo de elementos
* total\_count: total de elementos registrados en el sistema que se pueden obtener, esto de conjunto con la cantidad de elementos limitados a mostrar se usa para calcular el paginado, paginas = total\_count / limit
* offset: Elemento a partir se devolvieron los datos
* limit: Limite a mostar solicitado

**Código HTTP:**

401 - Unauthorized

200 - Ok

Ejemplos de solicitud de información:

**GET**: https://gp.prod.uci.cu/users.json

por defecto si no se especifican parámetros de filtros generales se asume que limit (25) por configuración y offset (0)

{

 "users": [

 {

 "id": 84,

 "login": "AnaRita",

 "admin": false,

 "firstname": "Ana Rita",

 "lastname": "Poyeaux ",

 "mail": "anarita@uci.cu",

 "created\_on": "2021-09-24T20:40:25Z",

 "updated\_on": "2021-09-24T20:40:46Z",

 "last\_login\_on": "2021-10-04T16:40:06Z",

 "passwd\_changed\_on": "2021-09-24T20:40:25Z",

 "twofa\_scheme": null

 },

 {

 "id": 81,

 "login": "DCA",

 "admin": false,

 "firstname": "Damaris",

 "lastname": "Cruz",

 "mail": "damaris@uci.cu",

 "created\_on": "2021-09-24T19:44:36Z",

 "updated\_on": "2021-09-24T19:54:30Z",

 "last\_login\_on": "2021-09-29T18:29:20Z",

 "passwd\_changed\_on": "2021-09-24T19:44:36Z",

 "twofa\_scheme": null

 },

 ...,

 ],

 "total\_count": 58,

 "offset": 0,

 "limit": 25

}

**GET**: https://gp.prod.uci.cu/users.json?limit=10&offset=15

En este ejemplo, se solicita al como máximo 10 elementos (limit=10) a partir del elemento 15 (offset=15)

{

 "users": [

 {

 "id": 10,

 "login": "doropesa",

 "admin": false,

 "firstname": "Daisy",

 "lastname": "Oropesa Méndez",

 "mail": "doropesa@uci.cu",

 "created\_on": "2021-02-26T15:32:01Z",

 "updated\_on": "2021-09-15T12:08:16Z",

 "last\_login\_on": "2021-08-02T20:01:54Z",

 "passwd\_changed\_on": null,

 "twofa\_scheme": null

 },

 {

 "id": 76,

 "login": "dsocarras",

 "admin": false,

 "firstname": "Dunior",

 "lastname": "Socarras Benitez",

 "mail": "dsocarras@uci.cu",

 "created\_on": "2021-09-20T23:39:32Z",

 "updated\_on": "2021-09-20T23:39:32Z",

 "last\_login\_on": "2021-09-20T23:39:32Z",

 "passwd\_changed\_on": null,

 "twofa\_scheme": null

 },

 ...

 ],

 "total\_count": 58,

 "offset": 15,

 "limit": 10

}

**GET**: https://gp.prod.uci.cu/users.json?limit=10&offset=15&status=3

Se solicita al como máximo 10 elementos (limit=10) a partir del elemento 15 (offset=15)

{

 "users": [

 {

 "id": 32,

 "login": "yordan.decoro",

 "admin": false,

 "firstname": "Yordan",

 "lastname": "Decoro Esperón",

 "mail": "yordan.decoro@etecsa.cu",

 "created\_on": "2021-03-22T17:43:21Z",

 "updated\_on": "2021-09-15T12:10:07Z",

 "last\_login\_on": null,

 "passwd\_changed\_on": "2021-03-22T17:43:21Z",

 "twofa\_scheme": null

 }

 ],

 "total\_count": 16,

 "offset": 15,

 "limit": 10

}

En este caso solo devuelve 1 aunque hayan 16 porque se solicita a partir del elemento 15 (offset=15)

Estos son algunos pocos elementos que se ha podido mostrar aquí, pero pueden ampliar en otros espacios ya referenciados en esta guía.

# Guías para HTML

[**https://ftp.prod.uci.cu/pub/Videos/Cursos/HTML5/**](https://ftp.prod.uci.cu/pub/Videos/Cursos/HTML5/)

HTML es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc.

El HTML es un lenguaje de marcación de elementos para la creación de documentos hipertexto, muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida, pueda enfrentarse a la tarea de crear una web.

# Guías para CSS

[**https://ftp.prod.uci.cu/pub/Videos/Cursos/CSS3/**](https://ftp.prod.uci.cu/pub/Videos/Cursos/CSS3/)

CSS viene de las iniciales de "Cascading Style Sheets", que traducimos por "Hojas de estilo en cascada". Es el segundo lenguaje más esencial para crear páginas web. El primero sería HTML, con el que se define el contenido de la página. El segundo CSS, con el que se define la parte de la presentación, es decir, cómo deben mostrarse los elementos de la página, su posición, forma, espaciados, colores y en resumen, toda la parte estética.

CSS no tiene ningún sentido sin HTML, igual que a día de hoy HTML sin CSS también resultaría imposible, ya que la web ha evolucionado de tal forma que su aspecto es fundamental y el CSS es la única vía para conseguir personalizarlo.

## Partes principales del CSS

Para desarrollar con CSS tenemos que trabajar con una serie de elementos, mediante los cuales se declaran los estilos, básicamente éstos son los más importantes:

Selectores, mediante los cuales podemos especificar qué elementos de la página nos estamos refiriendo

Atributos de estilo para definir qué cosas queremos estilizar sobre los selectores indicados

Una serie de valores, que indican qué estilo se debe aplicar a cada atributo sobre cada selector. Los valores se expresan con unidades CSS, que sirven para cuantificar los valores (píxeles, puntos...)