

Desarrollo de un template responsive para publicidad bajo la tecnología Zurb Foundation

Fecha de recepción: 2020-07-01 • Fecha de aceptación: 2020-07-31 • Fecha de publicación: 2020-10-10

Luis Roberto Chica Armijos

Publipromueve S.A. Ecuador

lrobertochica@gmail.com

<https://orcid.org/0000-0002-1058-6928>

RESUMEN

La maquetación responsive de un boletín para campañas publicitarias resulta poco eficiente si se realiza con herramientas no estandarizadas o CMS (Sistemas de Gestor de contenido), estas son orientadas a usuarios sin conocimientos de html (HyperText Markup Language) y de uso libre limitado; o laboriosos si se utiliza en cambio lenguaje nativo html, lo cual aumenta la complejidad del desarrollo debido a la cantidad de etiquetas que se deben incluir en toda la estructura del código; todo ello sumado a las restricciones que presenta cada cliente de correo electrónico al renderizar en html, lo que ocasiona generalmente problemas en la visualización del boletín y aumento en la carga operativa del desarrollo.

Este trabajo propone utilizar la herramienta Zurb Foundation para el desarrollo de un template responsive, esto permite disminuir la carga operativa del desarrollador, facilitando la escritura de código y la visualización de los cambios en un navegador web en tiempo real, con la certeza de que el boletín se visualice correctamente en los principales dispositivos móviles y clientes de correo electrónico.

PALABRAS CLAVE: campañas publicitarias, Zurb Foundation, diseño adaptativo, boletín.

ABSTRACT

The responsive layout of a newsletter for advertising campaigns is not very efficient if it is done with non-standardized tools or CMS (Content Management Systems), these are oriented to users without knowledge of html (HyperText Markup Language) and limited free use; or laborious if native html language is used instead, which increases the complexity of the development due to the amount of tags that must be included in the entire structure of the code; all this added to the restrictions that each e-mail client presents when rendering in html, which generally causes problems in the visualization of the newsletter and increases the operational load of the development.

This work proposes to use the Zurb Foundation tool for the development of a responsive template, this allows to decrease the operational burden of the developer, facilitating the writing of code and the display of changes in a web browser in real time, with the certainty that the newsletter is displayed correctly on major mobile devices and email clients.

KEYWORDS: advertising campaigns, Zurb Foundation, adaptive design, newsletter.

Introducción

Actualmente el envío de boletines publicitarios se envía en formato no amigable como imágenes en JPG y PNG, lo que conlleva a que el usuario no pueda interactuar con la información del boletín, dificultando su correcta adaptabilidad y visibilidad por medio de los distintos dispositivos informáticos como *smartphone*, Pcs de escritorio y tablets. Esto generará un incremento de nuevos clientes, que a su vez mejorará la rentabilidad a la empresa.

Otro inconveniente es la visualización no responsive del boletín en los dispositivos móviles. En la siguiente *Figura 1* de Causa-Efecto de Ishikawa se describe este inconveniente.



Figura 1. Diagrama Causa-Efecto de Ishikawa, de envío de boletines no responsive

Fuente: elaboración propia

En relación al trabajo presentado se hace referencia a algunos trabajos realizados bajo la tecnología Zurb Foundation, que han tenido gran éxito en el mercado, tales como:

“Desenvolvimiento front-end de sistemas web no Armazém Paraíba” de Schulze (2018), template responsive para estrategia de marketin de analistas de correo electrónico. “Projeto de website para a 8ª Noite de Prêmios ESPM – Sul” de Azevedo (2014), proyecto para la creación de un sitio web responsive, bajo la tecnología Zurb Foundation e “Implementation of a Convenient School Content Management System” de Hudson (2015). proyecto para crear un sitio web en un colegio que permita la actualización de eventos y noticias, permitiendo la correcta información a los padres de familia y docentes.

Es importante destacar que Zurb Foundation es un framework que está en constante crecimiento y mejora, ya que desde su primer lanzamiento del 2011 ha ido ganando confianza en la comunidad de desarrolladores *front-end*, por las bondades indicadas anteriormente, adicionalmente que se integra con GitHub, Node.js, NPM (gestor de paquetes) y Visual Studio Code (VSC), componentes y herramientas de nuevas tecnologías de desarrollo (Get.foundation, 2020).

Zurb Foundation permite no solo desarrollar *template responsive* para email, sino que también sitios web y web apps; adicionalmente en GitHub y GitLab se puede encontrar proyectos en referente a *template* (boletín) para email (Emily, 2016), los cuales han tenido colaboraciones por

parte de la comunidad. Por lo que, de acuerdo a los proyectos mencionados anteriormente, se expone el presente trabajo que se describe a continuación.

Con el *template* se pretende receptor usuarios para que puedan interactuar con la información contenida dentro del boletín. También se espera facilitar el historial de búsquedas de boletines por alguna palabra o texto relevante que se haya incluido dentro del mismo, ya que muchas ocasiones se envía tan solo una imagen publicitaria, dificultando su ubicación posterior. Igualmente, la idea es facilitar la reutilización del *template*, siendo únicamente necesario el reemplazo de nuevas imágenes para las distintas campañas publicitarias que se vayan requiriendo. Adicionalmente, se espera reutilizar el código, especialmente en tablas con datos numéricos, reduciendo de esta forma la carga operativa para la entrega de nuevos boletines.

En cuanto al alcance del desarrollo será un *template responsive* bajo Zurb Foundation, que tiene las siguientes características:

- **Responsive:** “Técnica de diseño web adaptable que busca la correcta visualización de una página o *template* en diferentes dispositivos como tablets, smartphone, libros electrónicos, portátiles, PC.” (Horev, 2014).
- **Header:** “Elemento html que provee información introductoria (títulos, subtítulos, logos)” (Barzanallana, 2015).
- **Body:** “Es el cuerpo o parte principal de la estructura de un documento HTML” (Gauchat, 2012).
- **Footer:** “Representa el final del cuerpo o de varias secciones de la estructura de un documento HTML” (Gauchat, 2012).
- **Texto informativo:** Texto agregado al boletín en formato editable (no imagen), para una mejor interacción con el usuario.

Metodología

Para la metodología de investigación se utilizó el método hipotético – deductivo, por cuanto parte de la observación del fenómeno a estudiar, creándose una hipótesis para explicar dicho fenómeno, deduciendo consecuencias o proposiciones más elementales de la propia hipótesis y por último permitiendo verificar y comprobar la verdad de los enunciados, comparándolos con la experimentación a partir de una campaña publicitaria con lo que se pretende mejorar su presentación e interacción del usuario, con el objetivo de captar clientes mediante un *template* amigable. En cuanto al método de desarrollo se utilizó eXtreme Programming (XP) (Joskowicz, 2008), debido a los constantes cambios solicitados por el cliente, XP permitió sobrellevarla, facilitando entregar pequeñas versiones o avances del software sobre cada iteración.

A continuación, se describe la tabla de iteraciones del plan de entregas (*Tabla 1*) de las 9 historias de usuario.

Tabla 1.
Plan de Entregas

Plan de Entregas del Template						
Nro. Historia	Nro. de Iteración	Desarrollo Módulos	Prioridad	Esfuerzo Estimado (Días)	Comienzo	Fin
1	1	Configuración de maqueta	alta	0	15/05/2020	15/05/2020
2	2	Header	alta	1	15/05/2020	15/05/2020
3	3	Banner	alta	2	21/05/2020	22/05/2020
4	4	Name	alta	1	27/05/2020	27/05/2020
5	5	Table-millas (saldo de millas)	alta	2	28/05/2020	29/05/2020
6	6	Contenido_1 y Contenido_2	alta	3	11/06/2020	13/06/2020
7	7	Footer (términos & condiciones)	alta	1	25/06/2020	25/06/2020
8	8	Template para campañas	alta	0	25/06/2020	25/06/2020
9	9	Visualización de pruebas de template	alta	0	25/06/2020	25/06/2020

Fuente: elaboración propia

Resultados

En cuanto a la obtención de resultados se usó las tarjetas CRC y las pruebas de aceptación del cliente, igualmente se detallan los estándares de programación utilizados para el desarrollo del template:

1. Tarjetas CRC

Luego de obtener las historias de usuario, se vio la necesidad de utilizar las tarjetas CRC, "(Clase, Responsabilidad y Colaboradores), ya que ayudan a identificar las responsabilidades y colaboradores de cada clase, y como se relacionan entre sí" (Escribano, 2002).

La de aprobación permite identificar los colaboradores responsables para la aprobación del template (ver *Tabla 2*).

Tabla 2.
Tarjeta CRC Aprobación

Aprobación	
Responsabilidades	Colaboradores
Proveer material para la actualización del template	Comunicación
Validar y aprobar el template responsive	
Recibir y visualizar el template responsive de la campaña publicitaria	Cliente de Base de Datos
Actualizar nuevo contenido del template, de acuerdo a la campaña vigente.	Usuario Técnico
Validar pruebas del template responsive	

Fuente: elaboración propia

2. Estándares de programación utilizados

Las herramientas y tecnologías que se utilizaron en el desarrollo del template son las siguientes:

- **CamelCase:** “recomendado como buena práctica para facilitar la lectura de código, tales como variables, clases, métodos o funciones, facilitando el mantenimiento del programa” Miller, F. P., Vandome, A. F., & McBrewster, J, (2009). Ejemplo: backColor, claseHeader, tablaFooter, etc.
- **Inky:** “es un lenguaje de plantillas que convierte etiquetas HTML simples en la tabla HTML compleja requerida para los correos electrónicos” (Foundation, 2018).
- **Css:** “es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, etc.” (Gauchat, 2012).
- **Sass:** “es una extensión de CSS que añade características muy potentes y elegantes a este lenguaje de estilos. Sass permite el uso de variables, reglas CSS anidadas, mixins, importación de hojas de estilos y muchas otras características, al tiempo que mantiene la compatibilidad con CSS” (Shenoy, 2016).
- **Html:** “es un lenguaje de marcas de hipertexto (HyperText Markup Language) que permite crear la estructura básica de páginas web, organizar su contenido y compartir información” (Giralt, 2011).
- **Node:** “es un entorno de ejecución de código abierto para JavaScript construido con el motor de JavaScript V8 de Chrom, permite trabajar con frameworks como zurb foundation, express, etc” (Node.js, 2020).

Por otra parte, para Cortizo Pérez, J. C., Expósito Gil, D., & Ruiz Leyva, D (2012) mediante las pruebas de aceptación “se da cumplimiento al plan de entregas por cada iteración, en la cual el

cliente valida el cumplimiento de cada historia de usuario”.

En la siguiente *Tabla 3* se describe la prueba de aceptación, para la historia de usuario 9.

Tabla 3.

Pruebas de Aceptación Visualización de pruebas de template

CASO DE PRUEBA DE ACEPTACIÓN	
Código: 9	Nro. Historia de Usuario: 9
Nombre: Visualización de pruebas de template	
Descripción: La visualización de las pruebas del template en los principales dispositivos, serán similares a las imágenes JPG originales tanto en formato desktop como responsive.	
Condiciones de Ejecución: Disponer de las pruebas del template en un solo documento PDF.	
Entrada/Pasos de ejecución:	
El usuario (cliente) visualiza todas pruebas del template en un solo documento consolidado.	
Resultado Esperado: El usuario (cliente) visualiza las pruebas del template satisfactoriamente.	
Evaluación de la Prueba: Exitoso	

Fuente: elaboración propia

Para el desarrollo de la propuesta se utilizaron las siguientes herramientas:

- Framework Zurb Foundation, versión 2.2.5 para maquetación responsive del template.
- Editor de código Visual Studio Code, versión 1.47.2 para entorno de edición de código.
- Navegadores web Google Chrome, Mozilla Firefox, Internet Explorer, version 84.0, 78.02, 11 respectivamente, para ver los cambios del código en el navegador web en tiempo real.
- Adobe Fireworks CS6 v12.0.0, para extraer los recortes de imágenes de la gráfica original.

A continuación, se indica el esquema de diseño la interfaz del template, de acuerdo a una base actual de imágenes para desplegar publicidad, se tomarán dichas imágenes o forma de despliegue y se realizará una adaptación de la propuesta realizada, por lo que a partir del diseño existente “Gráfica Original” (*Figura 2*) se realizará un análisis de sus componentes mostrados en la gráfica “Esquema de diseño de la interfaz de usuario” (*Figura 3*) para una posterior implementación de la propuesta.

Diseño Desktop

Diseño Responsive

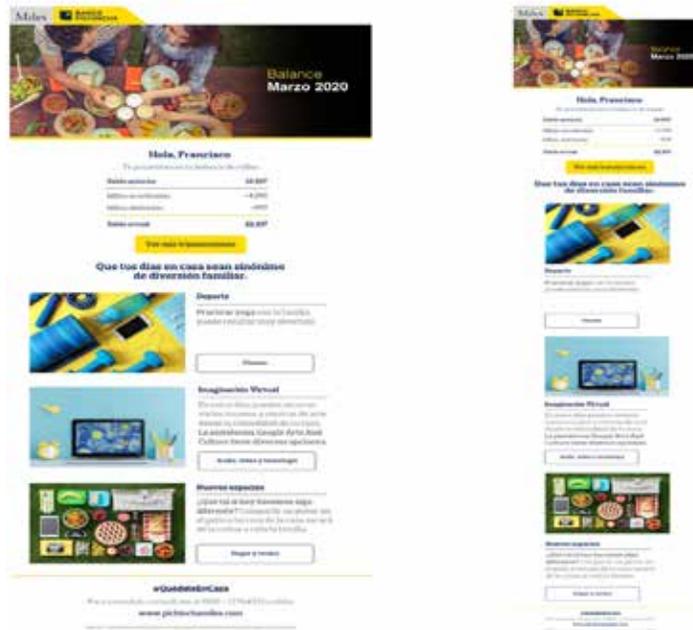


Figura 2. Gráfica original
Fuente: Publipromueve S.A

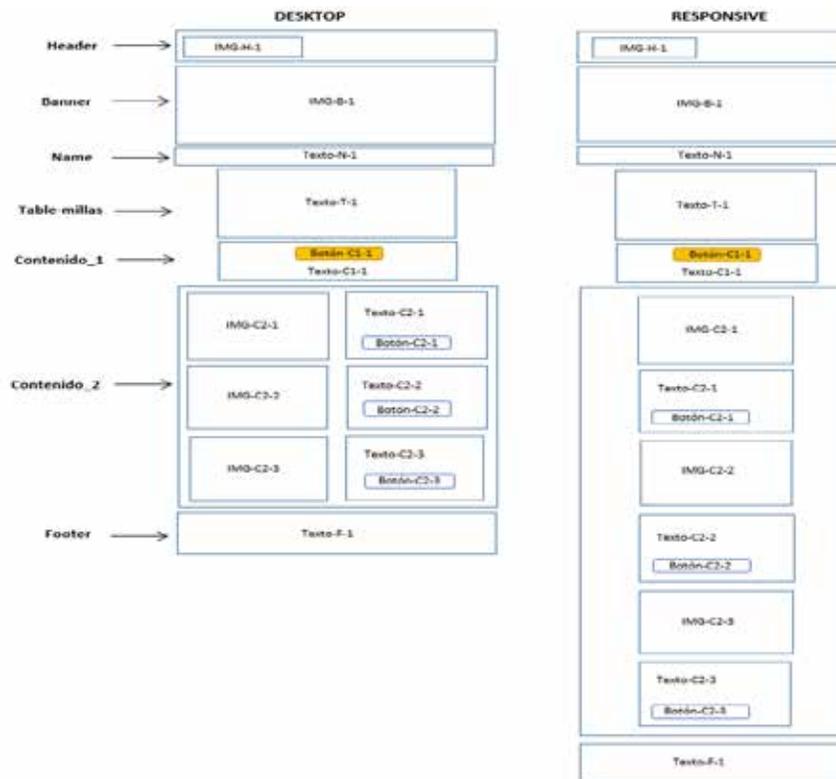


Figura 3. Esquema de diseño de la interfaz de usuario
Fuente: elaboración propia

En consiguiente se evidencian las pruebas realizadas mediante las herramientas de marketing digital Teradata-DMC (*Figura 4*) y Mailchimp (*Figura 5*), las cuales son aplicaciones web de pago para envíos masivos a grandes cantidades de contactos, a la vez que permiten emular las pruebas de boletines html en los principales dispositivos móviles y clientes de correo electrónico como Android, iPhone, iPad, Outlook, Gmail, etc.

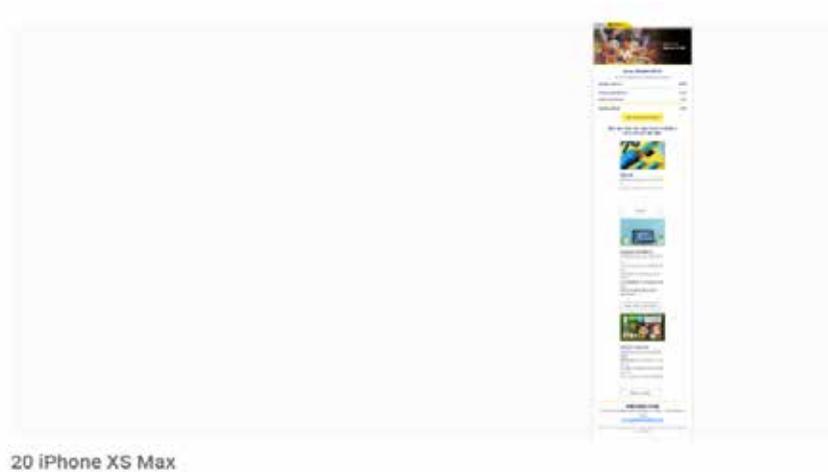


Figura 4. Visualización en iPhone XS Max con Teradata-DMC

Fuente: elaboración propia

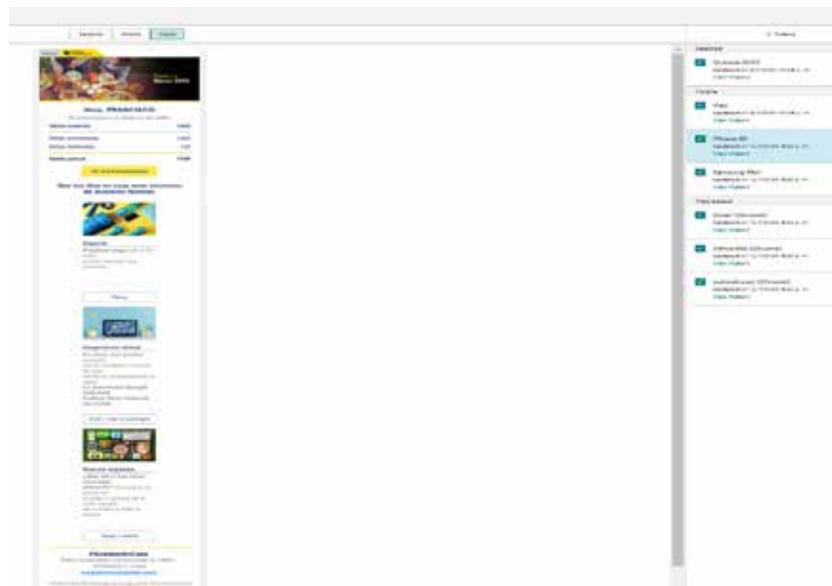


Figura 5. Visualización en iPhone 6S con Mailchimp

Fuente: elaboración propia

En la siguiente *Figura 6* se muestra como sería la visualización en Gmail con ingreso en Chrome.

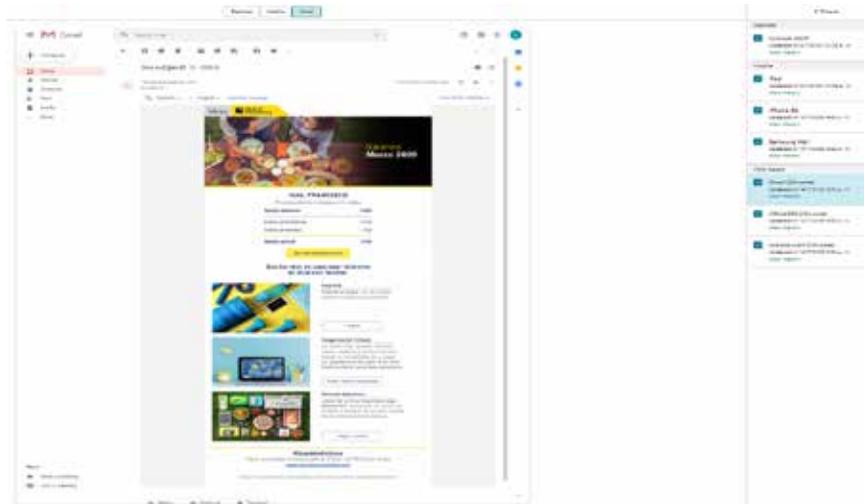


Figura 6. Visualización en Gmail (Chrome)

Fuente: elaboración propia

Para la implementación del proyecto se utilizará Google Drive como repositorio de imágenes y Sendinblue como herramienta de envío del template hacia el destinatario, y principalmente el archivo .html (template) generado por Zurb Foundation con el comando “npm run build”. Es importante guiarnos por la documentación de ayuda sobre los comandos y componentes que dispone el framework, la cual podemos obtener desde el repositorio de Zurb Foundation.

Una vez ejecutado el comando se levantará el proyecto en el que se podrá visualizar el template en el navegador web, verificando de esta forma su adaptación tanto en desktop (Figura 7), como responsive (Figura 8).

**Proyecto Levantado/Ejecutado
(Vista Desktop)**

Entorno de desarrollo

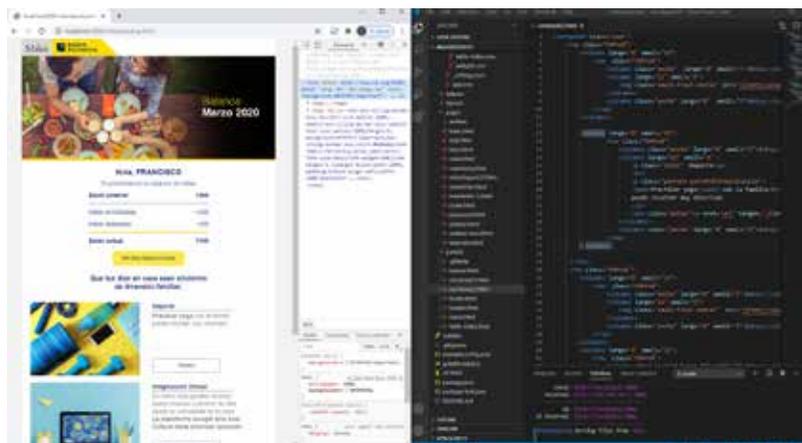


Figura 7. Visualización en tiempo real del template levantado en Zurb Foundation (Vista Desktop))

Fuente: elaboración propia

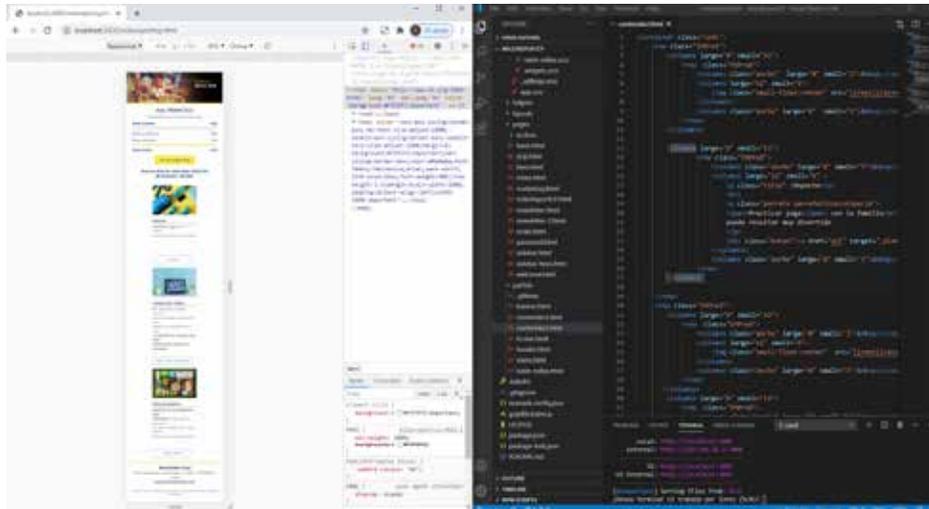


Figura 8. Visualización en tiempo real del template levantado en Zurb Foundation (Vista Responsive))

Fuente: elaboración propia

Se puede verificar que seleccionando el template tanto en desktop, como responsive, se encuentra incluido el texto, imágenes y botones, lo cual permitirá una mejor interacción con el usuario (Figura 9).

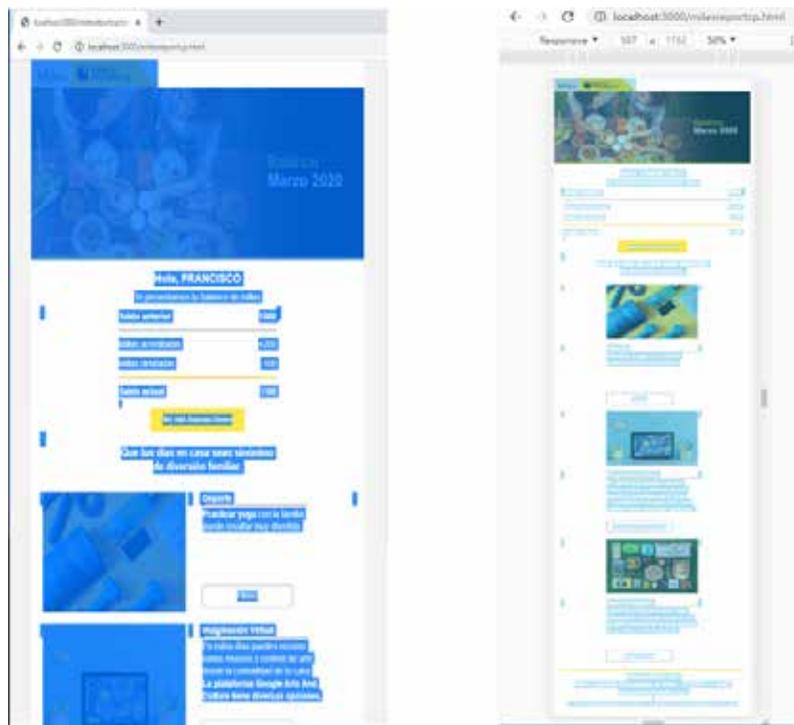


Figura 9. Verificación de contenido del template levantado en Zurb Foundation

Fuente: elaboración propia

Luego deberemos obtener el archivo .html, el cual se ubica en el directorio: Documents \ foundation-proyect\milesReportCP\dist\milesReportCP.html, de igual forma las imágenes que se encuentran en el directorio (Figura 10 y Figura 11) ...\Documents\foundation-proyect\ milesReportCP\src\assets\img\images deberemos subirlas al repositorio de Google Drive (Figura 12).

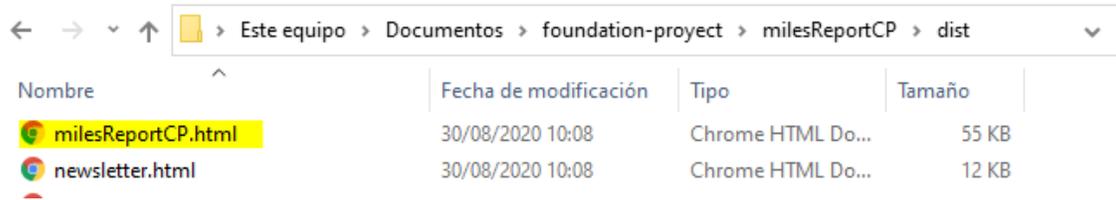


Figura 10. Archivo .html (template) generado por Zurb Foundation

Fuente: elaboración propia

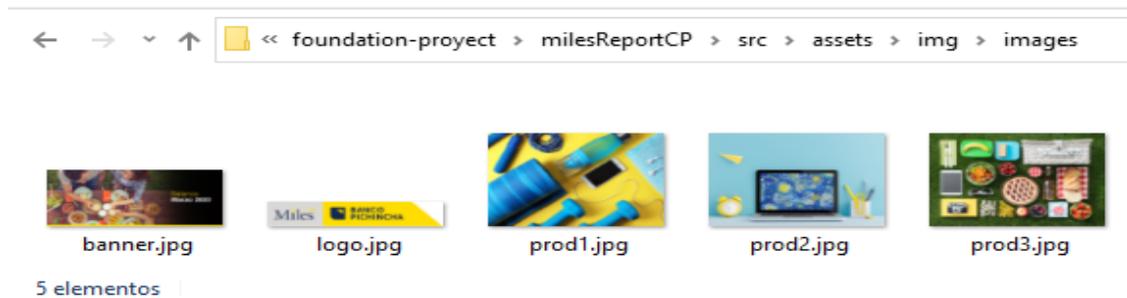


Figura 11. Imágenes del template

Fuente: elaboración propia

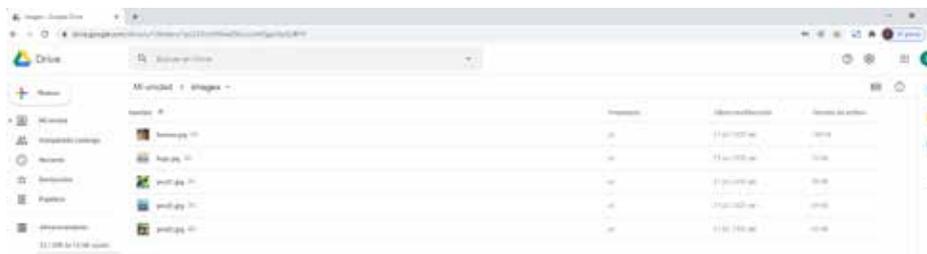


Figura 12. Imágenes subidas en el repositorio Google Drive

Fuente: elaboración propia

Luego de esto, deberemos subir el template en SendinBlue <https://app.sendinblue.com/account/login>, el acceso se realizará con una cuenta gratuita para efectos demostrativos (Figura 13).

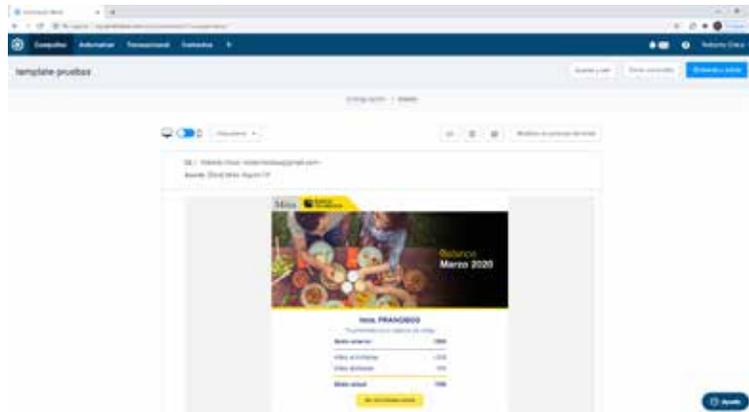


Figura 13. Template subido en Sendinblue

Fuente: elaboración propia

Finalmente enviaremos una prueba (*Figura 14*) a una cuenta de correo electrónico para su recepción por parte del cliente.

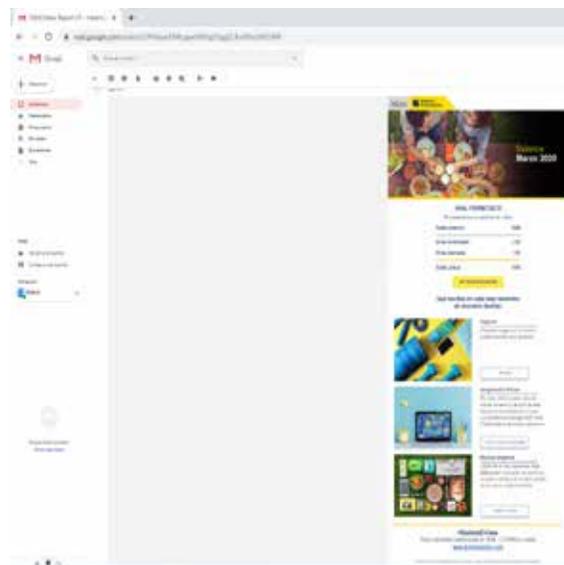


Figura 14. Recepción de prueba en el cliente de correo electrónico Gmail

Fuente: elaboración propia

En cuanto a la demostración de recepción en el dispositivo móvil se utilizó la aplicación Conectar de Window 10 (*Figura 15*).

Las diferentes secuencias de imagen son los deslizamientos realizados en el dispositivo para de esta forma visualizar completamente el template.

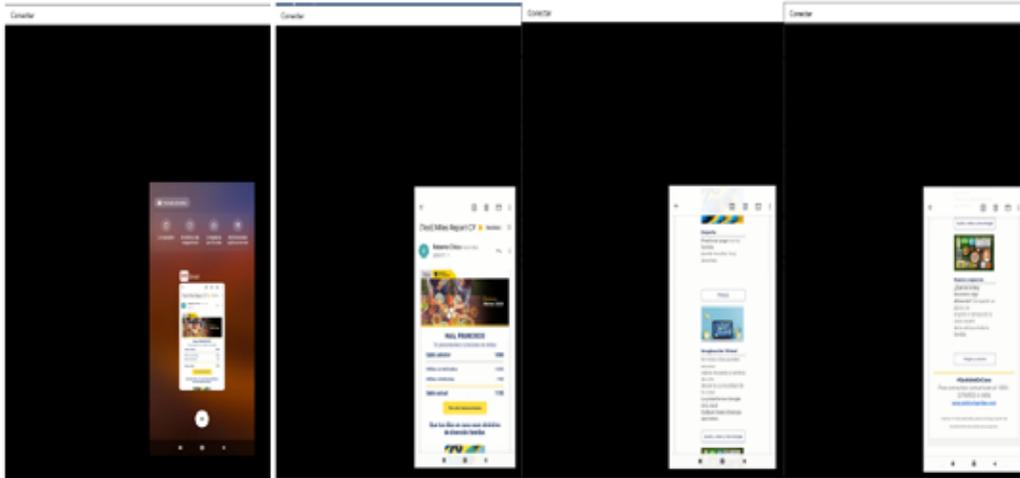


Figura 15. Recepción de prueba en el cliente de correo electrónico Gmail en un dispositivo móvil
Fuente: elaboración propia

Mientras que para la experimentación se utilizó Stripo.email (Figura 16) como herramienta CMS (Sistemas de Gestor de contenido), ya que este permite crear *templates* rápidamente con la función arrastrar y soltar, sin necesidad de incluir código html; pero en cambio se obtiene un *template* el cual no se visualiza correctamente su contenido como texto (tabla de millas y sus líneas separadoras) en los principales clientes de correo como Outlook.com (Figura 17).

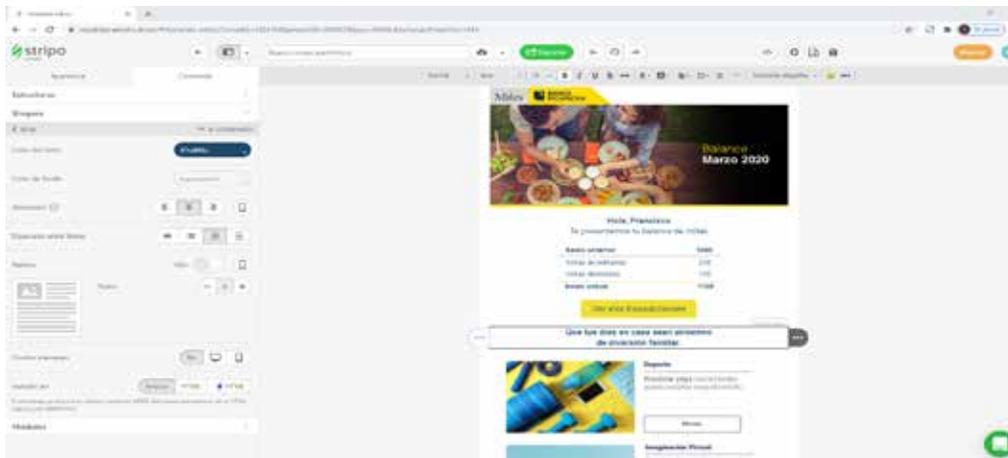


Figura 16. Diseño de template en Stripo.email
Fuente: elaboración propia

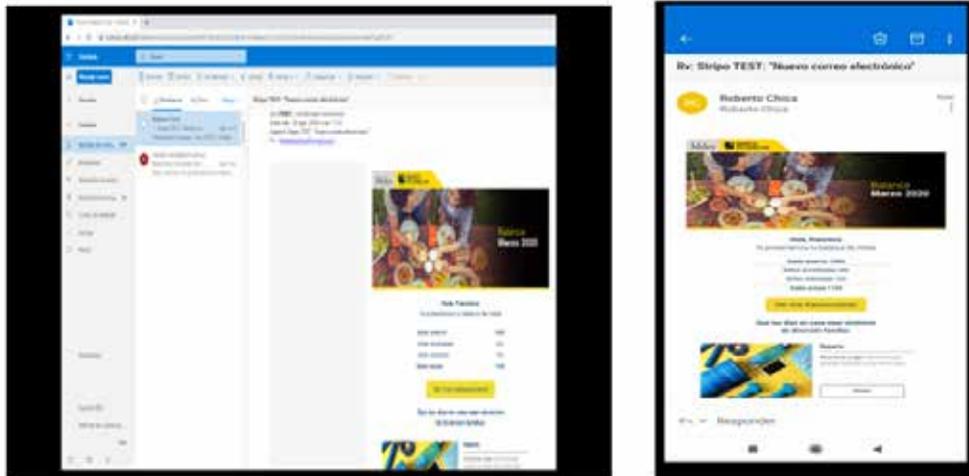


Figura 17. Pruebas de template en cliente de correo electrónico Outlook.com

Fuente: elaboración propia

Adicional a lo expuesto, se detallan otros inconvenientes presentados:

- En las pruebas de visualización mediante navegadores web, especialmente Internet Explorer (Figura 18), no se visualiza correctamente en las diferentes versiones (10, 9, 8 y 5), a diferencia del template desarrollado en Zurb Foundation (Figura 19); lo cual es muy importante, debido a que de esta forma se comprobará la visualización correctamente en estas versiones del navegador y de igual forma en el cliente de correo Outlook de Microsoft.

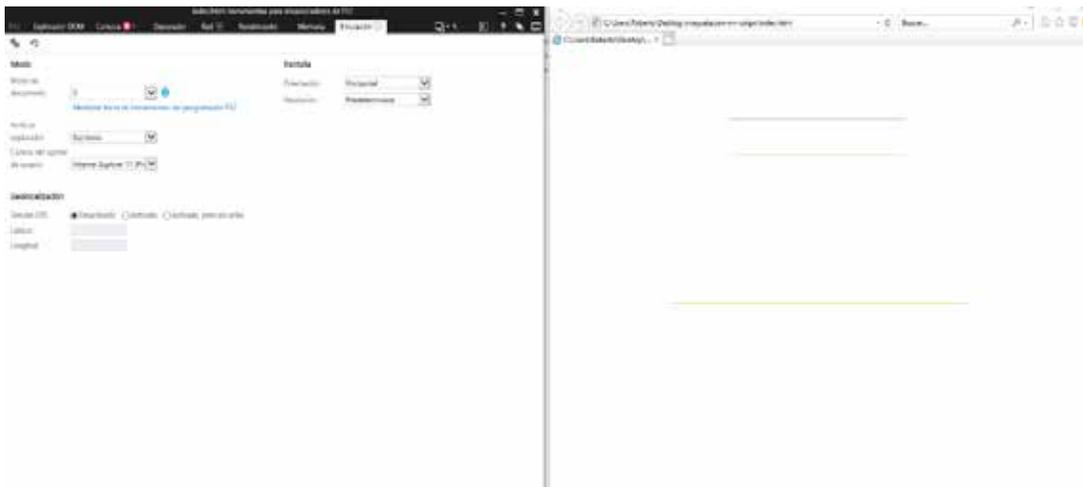


Figura 18. Pruebas de visualización en Internet Explorer 5 (Template Stripo.email)

Fuente: elaboración propia

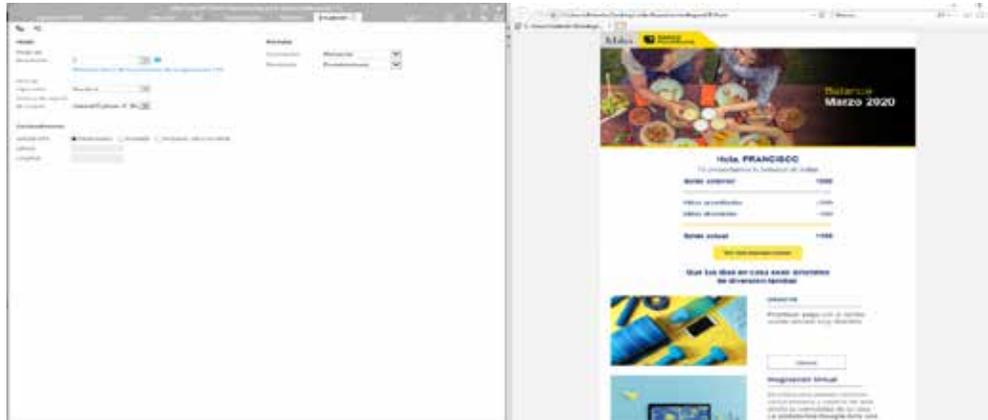


Figura 19. Pruebas de visualización en Internet Explorer 5 (Template Zurb Foundation)

Fuente: elaboración propia

- No se pueden realizar pruebas con el *template* obtenido con herramientas de emulación (Mailchimp) en los principales dispositivos, debido a que se genera una etiqueta de Java Script (Figura 20), lo cual restringe la carga y subida del *template*, si se elimina esta etiqueta en cambio las imágenes no se visualizan, por lo que, solo se podrá enviar únicamente por el aplicativo Stripo.email.

```

1  <!doctype html>
2  <html <!-- email -->
3  <head><meta charset="utf-8"><style ampemail-boilerplate>body{visibility:hidden}</style> <script async src="https://cdn.ampproject.org/v0.js"></script>
4
5  <style amp-custom>
6  @media only screen and (max-width:600px) {p, ul li, ol li, a { font-size:150%; line-height:150% } h1 { font-size:300%; text-align:center; line-height:120% } h2 { font-size:250%;
7  a[x-ampdata-detectors] {
8  color:inherit;
9  text-decoration:none;
10 font-size:inherit;
11 font-family:inherit;
12 font-weight:inherit;
13 line-height:inherit;
14 }

```

Figura 20. Etiqueta de Java Script incluida en template generado por Stripo.email

Fuente: elaboración propia

De igual forma, no se puede implementar el template por Sendinblue.com (Figura 21), debido a las restricciones indicadas anteriormente.

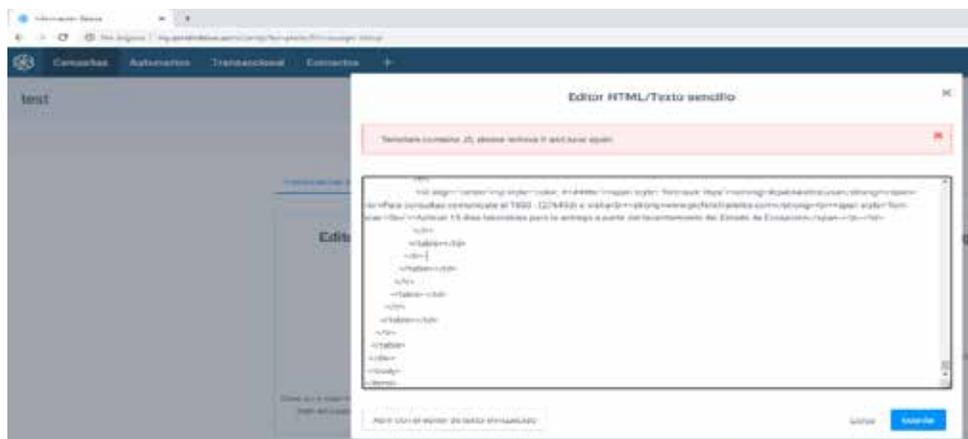


Figura 21. Restricciones de carga de template en Sendinblue

Fuente: elaboración propia

El envío de pruebas y descarga del template en la versión gratuita está limitado a 5 veces diarias, si se sobrepasa esta cantidad se deberá esperar su reinicio al siguiente día.

Para la siguiente experimentación en lo referente a la maquetación tradicional, en la que el *template* no será responsive (*Figura 22*), como se indicó al inicio, este requerimiento dependerá del cliente, por tanto, solo se hará uso del editor de código Visual Studio Code (VSC) y Adobe Fireworks CS6 para obtener la estructura html principal del *template* y las imágenes de la gráfica, con lo que solo será necesario personalizar los campos como nombres, agregar estilos CSS básicos y los links correspondientes a las imágenes.

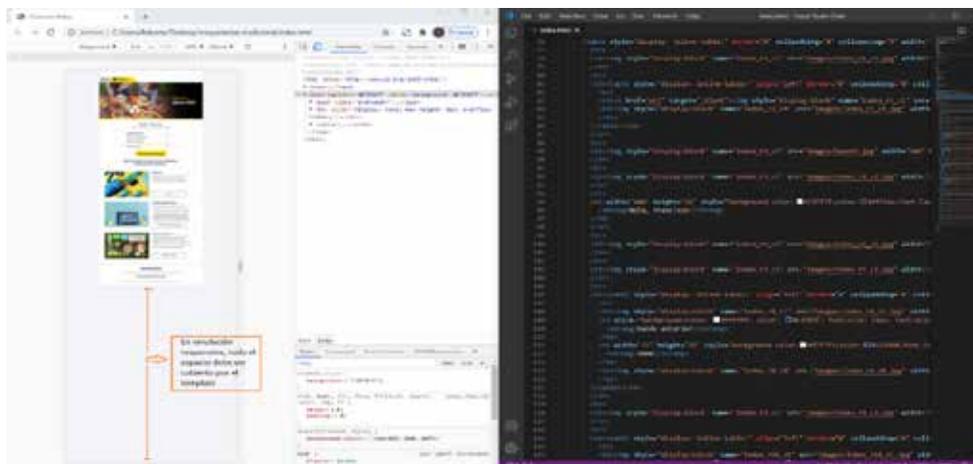


Figura 22. Visualización no responsive del template en Google Chrome

Fuente: elaboración propia

Como se puede observar en la siguiente *Figura 23* las imágenes de acuerdo al “Esquema de interfaz de usuario” mantienen la misma ubicación, por lo que en un diseño responsive se busca

su reubicación y adaptabilidad de acuerdo a las preferencias del cliente para mejorar la visibilidad de las imágenes y que las mismas no pierdan su resolución y nitidez en los diferentes dispositivos móviles, de la misma forma para las secciones de imagen con texto es necesario incluir como texto más, no como imágenes, de acuerdo a lo indicado al inicio en la justificación del desarrollo del template responsive.

Vista en Dispositivo Desktop

Vista en Dispositivo Móvil

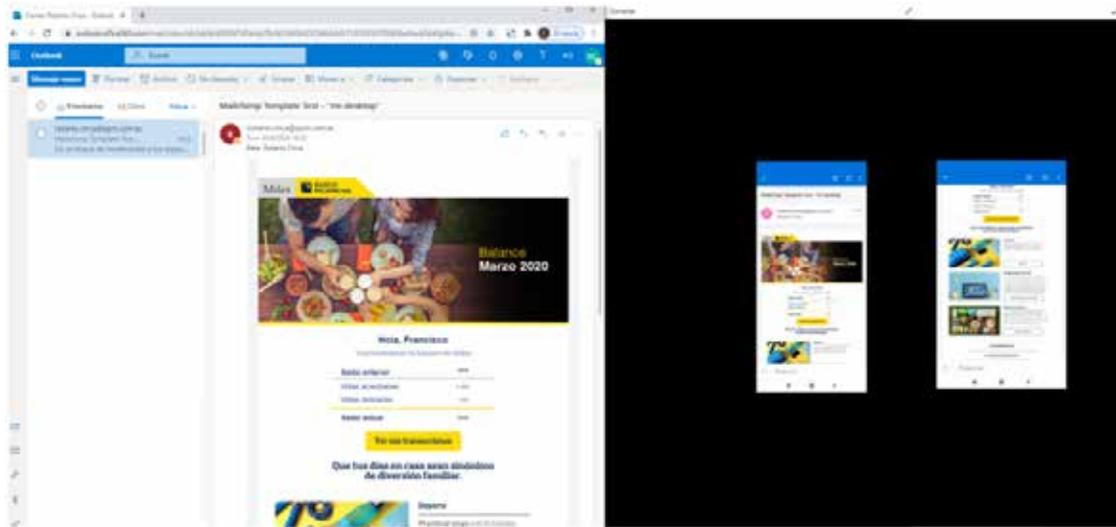


Figura 23. Pruebas de template no responsive recibidas en cliente de correo electrónico Outlook.com

Fuente: elaboración propia

Conclusiones

Se pudo comprobar que el hacer uso de herramientas, tanto en la metodología XP, como el framework Zurb Foundation, facilitaron el avance para el cumplimiento del ciclo de desarrollo del proyecto como son: Planificación, desarrollo, integración, pruebas y entregables.

Adicionalmente, se pudo evidenciar la utilidad de contar con una herramienta de desarrollo para obtener un *template* (boletín) compatible con la mayoría de clientes de correo electrónico, optimizando su renderización (convertir el código html para representarlo visualmente).

El desarrollo del *template* para publicidad bajo la tecnología Zurb Foundation permitió agilizar la entrega de nuevos boletines, contribuyendo al cumplimiento de los objetivos y mejorar la productividad de la empresa.

Se observó a su vez que el framework Zurb Foundation facilitó la escritura de código gracias a sus dos componentes principales como son el lenguaje de plantilla Inky que permite agilizar la escritura de código html, y el visualizador de template BrowserSync, que permite ver los cambios del código en el navegador web en tiempo real.

Se pudo demostrar también la importancia de utilizar los artefactos de la metodología ágil XP, y como estos ayudaron al cumplimiento al ciclo de vida del proyecto, ya que permitió una adecuada integración de nuevos cambios solicitados por el cliente, sobre la marcha del desarrollo del proyecto; facilitando entregar pequeñas versiones o avances del software sobre cada iteración.



Referencias

- Azevedo, G. (2014). Projeto de website para a 8ª Noite de Prêmios ESPM - Sul. XV Congresso de Ciências da Comunicação na Região Sul – Palhoça <https://www.portalintercom.org.br/anais/sul2014/expocom/EX40-0411-1.pdf>
- Barzanallana, R. M. (2015). Lenguaje de programacion HTML y CSS. <https://www.docsity.com/es/lenguaje-de-programacion-html-1/3284771/>
- Cortizo Pérez, J. C., Expósito Gil, D., & Ruiz Leyva, D. (2012). eXtreme Programming.
- Emily. (2016). Why Clients Render Email Differently. Mailchimp. <https://mailchimp.com/resources/why-clients-render-email-differently/>
- Escribano, G. F. (2002). Introducción a Extreme Programming. Ingeniería del Software, 11.
- Foundation, Z. (2018). Inky. <https://get.foundation/emails/docs/inky.html>
- Gauchat, J. D. (2012). El gran libro HTML5, CSS3 y Javascript. <https://gutl.jovenclub.cu/wp-content/uploads/2013/10/EI+gran+libro+de+HTML5+CSS3+y+Javascrip.pdf>
- Get.foundation. (2020). Zurb Foundation. <https://get.foundation/>
- Giralt, M. L. (2011). Introducción al HTML y al CSS.
- Horev, K. (2014). Learning Zurb Foundation. birmingham: Packt Publishing Ltd.
- Hudson, J. (2015). Implementation of a Convenient School.
- Joskowicz, J. (2008). Reglas y Prácticas en eXtreme Programming.
- Miller, F. P., Vandome, A. F., & McBrewster, J. (2009). CamelCase: Compound (linguistics), Whitespace (computer science), Capitalization, Patti LaBelle, Visual Basic, MacGyver, iPod, Chemical formula, Naming... Programming language, Marketing.
- Node.js. (2020). About Node.js®. <https://nodejs.org/en/about/>
- Schulze, L. A. (2018). RELATÓRIO DE ESTÁGIO. https://estudante.ifpb.edu.br/media/cursos/39/documentos/TCC_Lucas_Schulze_-_Final.pdf
- Shenoy, A. (2016). Introducing Zurb Foundation 6. Apress.