



**bizagi**

# Prueba Técnica de Desarrollo

## Frontend Developers

## Tabla de Contenidos

Introducción.....	2
Requerimientos Funcionales.....	2
Requerimientos No Funcionales.....	3
Entregables .....	4
Evaluación.....	4

## Introducción

Usted acaba de ser contratado como ingeniero de desarrollo para Bizagi, la suite de BPM más popular del mundo. Mientras usted revisa el backlog de características nuevas para incorporar en el producto, encuentra que una de las características más pedidas por los usuarios es que el Inbox del WorkPortal pueda funcionar en browsers de múltiples dispositivos como Desktops, Tablets y Smartphones.

La función principal del Workportal de Bizagi es para presentar al usuario final las actividades tiene pendientes en los procesos en que está participando y permitirle trabajar en ellas. El Inbox es una vista que permite al usuario final ver las actividades que tiene pendiente, elegir una de ellas, y diligenciarla.

Usted decide desarrollar esta funcionalidad para complacer las necesidades de los usuarios de Bizagi así que inicia una investigación de los requerimientos con más detalle.

## Requerimientos Funcionales

Para limitar el problema, el proceso sobre el cual se va a trabajar es un proceso de vacaciones y vamos a suponer que el usuario es el jefe del área y por lo tanto solo se van a manejar actividades de aprobación o rechazo de vacaciones.

El usuario final puede:

1. Ver una lista de sus actividades pendientes (Inbox)
  - a. El usuario debe poder ver la información básica de cada actividad, la que usted considere, por ejemplo, quien solicita las vacaciones y cuántos días está solicitando
2. Ver el detalle de una actividad pendiente (al seleccionarla en el Inbox)
  - a. El usuario debe poder ver la información detallada de la actividad
  - b. El usuario debe poder ver el número de días calendario de vacaciones que se están solicitando
3. Aprobar o rechazar las vacaciones que le están solicitando en la actividad, una vez hecho esto, la actividad debe marcarse con un indicador visual en el Inbox que muestre si está aprobada o rechazada.

## Requerimientos No Funcionales

1. Tecnologías: Usted decide usar HTML, CSS y Javascript o jQuery para hacer este desarrollo. Usted decide también que la arquitectura sea MVC en el cliente y si lo ve necesario puede apoyarse en cualquier framework MVC en Javascript existente como Angular.js, Backbone.js o cualquier otro.
2. Estructura de archivos: Por razones de mantenimiento, usted decide no mezclar todo el código en los mismos archivos sino hacer una separación adecuada de los archivos de acuerdo a su criterio.
3. Usabilidad: Usted nota que la experiencia de usuario es uno de los factores de éxito más importante de Bizagi como producto, por lo tanto usted se traza como objetivo que esta característica mantenga un buen nivel de usabilidad.
4. Soporte a múltiples dispositivos: Dado que los usuarios necesitan poder acceder a su Inbox de actividades pendientes desde cualquier dispositivo, usted decide desarrollar la funcionalidad con responsive design.
5. Soporte a múltiples navegadores: Usted encuentra que los navegadores más usados por los usuarios de Bizagi son: Internet Explorer y Chrome y se traza como objetivo que la aplicación funcione al menos en estos dos navegadores.
6. Fluidez en la interacción: Usted decide que la interacción entre la aplicación cliente que va a desarrollar y el servidor debe hacerse vía Ajax para que el usuario final no perciba bloqueos o latencia mientras la está usando.
7. Formato de intercambio de datos: Usted considera que es importante ahorrar ancho de banda entre los dispositivos cliente y el servidor, así que decide que el formato de intercambio de datos entre la aplicación cliente y el servidor sea JSON.
8. Servidor: Para simular el servidor, use los archivos JSON que encontrará junto con este enunciado: inbox.json (Contiene el listado de actividades pendientes para el usuario).

## Entregables

Usted debe entregar el código fuente de la aplicación y si lo desea puede publicar la aplicación funcionando en algún servidor de su preferencia.

## Evaluación

Los criterios de evaluación son los siguientes:

1. El correcto funcionamiento de la aplicación.
2. La arquitectura y diseño plasmado en su solución.
3. Estructura y organización de la solución.
4. Estándar de codificación.
5. Capacidad algorítmica.
6. Uso de la tecnología.