

# Producción

- Proceso de Despliegue:
  - No existe ningún proceso documentado de cómo realizaremos los despliegues. Algunos conceptos relevantes.
    - CI CD
    - Git Pull Request
    - Git Branches
    - QA, Unit Testing y Integrations Testing
- Certificados Https:
  - Estamos usando los certificados gratuitos de https, esto en sí no es un problema pero fue una solución parcial que realicé en su momento para sacarnos del apuro, un SAAS de entel debería contar con sus propios y oficiales certificados.
- Pruebas Automatizadas:
  - Todo lo relacionado a la validación correcta de funcionalidades nuevas y existentes, Unit Testing e Integration Testing probablemente son los más relevantes.
  - Mi idea es integrar el sistema con Github Actions para que al momento de hacer un pull request se corran automáticamente las pruebas predefinidas, además de hacer una build de prueba, esto obviamente no es perfecto pero es mucho mejor que lo actual.
    - Para Express: Mocha & Chai son lo suficiente para describir y validar resultados válidos, como preferencia Axios para request,
    - Para Laravel: PHPUnit.
- Repositorios:
  - Seguimos usando el github de ivan como repositorio principal, todo el CI/CD que se quiera construir quedará asociado a su cuenta.
  - Optimizaciones en el monorepo
- Entorno de pruebas:
  - Se requiere configurar por completo QA para que se adapte a los nuevos sistemas.
  - Se requieren más dominios para las nuevas rutas.
- Emails:
  - Considerar reputación de dominio, posible recepción como spam
  - Validar correcta recepción de correos en mobile, outlook, emails de apple, etc.
  - ¿Cuántos emails mensuales tenemos?
  - Entorno de qa para correos
  - Escalabilidad y trazabilidad del servicio actual

# Continuidad Operativa

- Sólo un servidor:
  - En caso de emergencia no tenemos a donde migrar
  - No hay plan ni proceso de migración
  - No hay tiempo de respuesta operativo
- Respaldos:
  - No tenemos un sistema real de respaldos.
  - No hay plan de migración y ni recuperación de respaldos.
    - ¿Qué hacer si se requiere la data de ayer? ¿Levantamos otro proyecto con la db? Se extrae en un entorno local? ¿Quién hace esta tarea?
- Base de Datos:
  - Actualmente tenemos 4 bases de datos, sólo sdx cuenta con respaldos.
  - Las bases de datos se encuentran en el mismo servidor de prod con el proyecto. Esto es un posible riesgo elevado
  - No hay plan de migración
  - La base de datos no está expuesta, nos conectamos por localhost, esto añade "algo" de seguridad, pero habrá que exponerla eventualmente al separarla de vm.
  - Seguimos usando phpmyadmin. Dependencia de php.
- Entorno:
  - El entorno actual relativamente sencillo de replicar pero toma tiempo.
    - Debian
    - Php 8.2
    - Python 3 + dependencias (wkhtmltopdf)
    - Node
    - Nginx
    - Dominios y Subdominios
  - Tiempo de recuperación o de crear un nuevo entorno no definidos.
- Media:
  - Actualmente almacenamos media en rutas dentro del proyecto de laravel, ejemplo: imagen de perfil  
<https://api.hubdesign.app/profile/ejemplo.jpg>
  - No hay respaldo de imágenes o archivos
  - No hay seguridad de imágenes o archivos
  - En caso de migrar el proyecto laravel a otra ruta se perderá todo
  - Vídeos o archivos pesados pueden usar todo el tráfico del servidor, al estar en el mismo server la carga también sube
- Alertas:

- No tenemos visibilidad del estado de nuestros sistemas a ningún nivel. Por ejemplo.
  - cpu sobre 90%
  - almacenamiento casi lleno
  - caídas
  - pérdidas de paquetes
  - ataque ddos
  - spam de request
  - error persistente en proyectos
- En caso de problemas la única forma que lo sepamos es probando nosotros.

## Tener servidores es un asco

- Respalos:
  - ¿Quién los revisa?
  - ¿Dónde se crean?
  - ¿Quién tiene acceso?
- Alertas:
  - Cpanel: Podría servir pero es bastante básico y no nos da vista a nivel proyecto del uso del sistema
  - Correos: No tengo idea.
- Almacenar Media:
  - Puedes usar un object storage como <https://min.io/> o crear otro servicio de express exponiendo una ruta pública pero...
    - Sistemas de Alertas o Eventos: Crear otro subsistema
    - Redundancia de servers.
    - Consistencia de urls y certificados https
    - Roles y acceso: GLHF
    - Límites de BW: GLHF
    - Mantenimiento: Todos los sistema sin media
    - Alertas de almacenamiento lleno?
    - Bandwidth?
    - Cors?
    - Seguridad del server vm
- Cantidad requerida de servidores:
  - Producción servidores:
    - 1. DB/MEDIA: Tampoco es buena idea dejar estos juntos:
      - Un cluster de Mysql con al menos tres nodos sería prudente.

- Lo mismo con min.io
- 2. Proyectos:
  - API express
  - API laravel
  - vue hubdesign
  - vue sdwan
  - vue sap
  - vue linkcloud
- QA mínimo 1 servidor: Todo lo anterior
- Servidor de respaldo:
  - Cuánto se demora entel en entregarnos otra instancia
  - ¿Qué tanto habrán cambiado nuestros entornos para cuando se requiera?
- No tenemos idea, mantener servidores con entel arruina el prototipado, el diseño y despliegue rápido, la arquitectura de un devops y la innovación.