**tTrainer**

**Sprint 3**

**Autor:**

Guillermo Fernández Pérez

3ºA GITT

Madrid, Enero 2020

Lista de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Descripción** | **Autor/es** |
| 0.1 | 02/01/21 | Primer borrador. | Guillermo Fernández Pérez |
| 0.2 | 08/01/21 | Primeros apartados. | Guillermo Fernández Pérez |
| 0.3 | 08/01/21 | Cambio de diagramas. | Guillermo Fernández Pérez |
| 1.0 | 10/01/21 | Revisión final. | Guillermo Fernández Pérez |

Índice

[1. Introducción - 5 -](#_Toc61190791)

[1 Retrosprectiva sprint 2 - 6 -](#_Toc61190792)

[2 Backlog del sprint 3 - 7 -](#_Toc61190793)

[3 Diseño del Proyecto - 8 -](#_Toc61190794)

[3.1 Modelos de Contexto - 8 -](#_Toc61190795)

[3.1.1 Modelo Arquitectónico - 8 -](#_Toc61190796)

[3.1.2 Modelo de Proceso Empresarial - 9 -](#_Toc61190797)

[3.2 Diseño de la Arquitectura Técnica - 10 -](#_Toc61190798)

[3.2.1 Aspectos Organizativos - 10 -](#_Toc61190799)

[3.2.2 Tecnología hardware a utilizar - 10 -](#_Toc61190800)

[3.2.3 Fronteras de mecanización - 11 -](#_Toc61190801)

[3.2.4 Plan de Fases restantes - 12 -](#_Toc61190802)

[3.3 Tabla de influencia de Requisitos - 12 -](#_Toc61190803)

[3.4 Diseño de Componentes - 13 -](#_Toc61190804)

[3.5 Diseño de Interfaz - 15 -](#_Toc61190805)

[3.5.1 Diagrama de navegabilidad - 15 -](#_Toc61190806)

[3.5.2 Diseño de pantalla - 16 -](#_Toc61190807)

[3.6 Diseño de base datos - 17 -](#_Toc61190808)

Lista de figuras

[Figure 1: Retrospectiva Sprint 2 - 6 -](#_Toc61190809)

[Figure 2: Planificación Sprint 3 - 7 -](#_Toc61190810)

[Figure 3: Diagrama de bloques - 8 -](#_Toc61190811)

[Figure 4: Modelo de proceso empresarial - 9 -](#_Toc61190812)

[Figure 5: Diseño hardware - 11 -](#_Toc61190813)

[Figure 6: Diagrama de Componentes - 14 -](#_Toc61190814)

[Figure 7: Diagrama de navegabilidad - 15 -](#_Toc61190815)

[Figure 8: Diseño de pantalla - 16 -](#_Toc61190816)

[Figure 9: Diagrama entidad relación - 17 -](#_Toc61190817)

Lista de TABLAS

[Table 1: Funciones de las capas de la arquitectura - 12 -](#_Toc61190818)

[Table 2: Tabla de influencia de requisitos - 13 -](#_Toc61190819)

# Introducción

En este documento se recoge toda la documentación generada en el tercer sprint del proyecto tTrainer.

En primer lugar se encontrarán las actividades relacionadas con la planificación:

* Retrospectiva del sprint 2
* Backlog del sprint 3

En segundo lugar se presentan varias tareas que se engloban dentro de las actividades del proceso de diseño del proyecto:

* Modelo de Contexto
* Diseño de la Arquitectura Técnica
* Tabla de influencia de requisitos
* Diseño de Componentes
* Diseño del Interfaz
* Diseño de la base de datos

Este segundo apartado se centra con más profundidad en la funcionalidad planificada hasta el tercer sprint, por lo que pueden quedar fuera alguna parte de la funcionalidad que aún no ha sido planificada o no se considera muy relevante para entender el contexto del proyecto.

# Retrosprectiva sprint 2

Para realizar esta retrospectiva se ha creado otro tablero Trello con los mismos apartados que en la anterior retrospectiva.

A continuación se presenta el tablero:

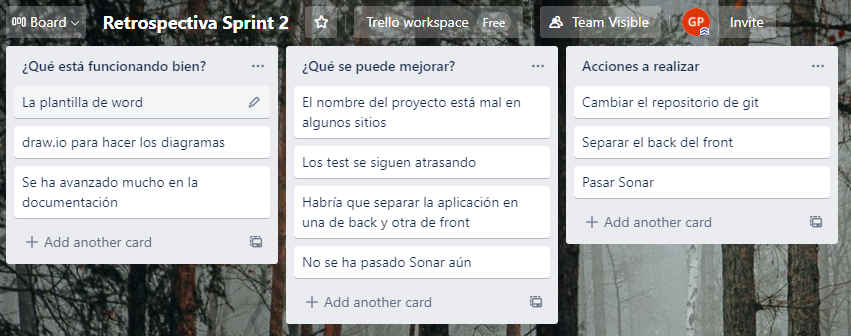


Figure 1: Retrospectiva Sprint 2

Se han identificado las siguientes mejoras:

* Cambiar el repositorio Git: Se llama iTranner y debería llamarse tTrainer.
* Separar el back del front: Ahora mismo todo está en una única aplicación, si se querría mejorar la interfaz, lo mejor sería tener la lógica en una aplicación y la interfaz en otra.
* Pasar el Sonar: La idea es ir encontrando errores de código, aunque el IDE identifica ya algunos.

# Backlog del sprint 3

A continuación se presenta el tablero al comienzo del tercer sprint:

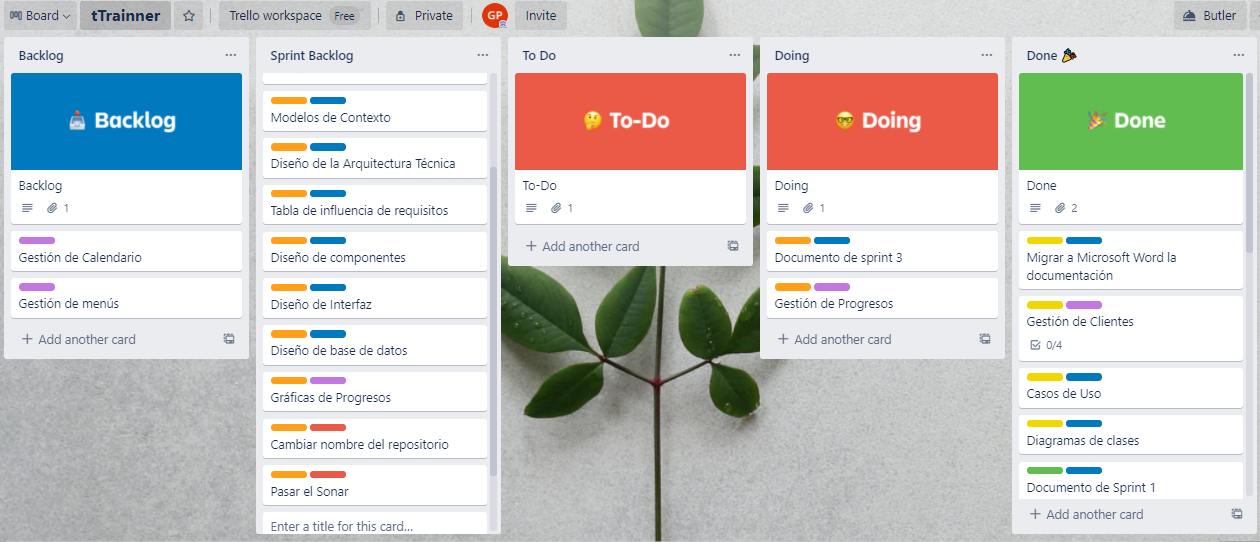


Figure 2: Planificación Sprint 3

Lo más representativo de este sprint es:

* Se ha añadido dos tareas de la retrospectiva.
* Las tareas de programación planificadas para este sprint están avanzadas.
* La documentación esta vez no se ha retrasado.
* Quedan dos tareas para el siguiente sprint, puede ser que no dé tiempo a finalizar alguna de ellas.

# Diseño del Proyecto

En este apartado se van a detallar diversas tareas de diseño del proyecto.

## Modelos de Contexto

En este apartado se enterará reflejar la arquitectura de la aplicación y su relación con otros sistemas y actores.

### Modelo Arquitectónico

En el siguiente diagrama de bloques se muestra la arquitectura básica del sistema. Como puntos a destacar tenemos:

* La conexión de los clientes con la aplicación va a ser a través de internet con http.
* Se incluye un balanceador para poder la aplicación en alta disponibilidad.
* Aunque actualmente la base de datos es una base de datos embebida en memoria, a la hora de desplegarlo en un entorno sólo habría que incluir un parámetro con la cadena de conexión. Por lo que la base de datos sería un servicio separado.

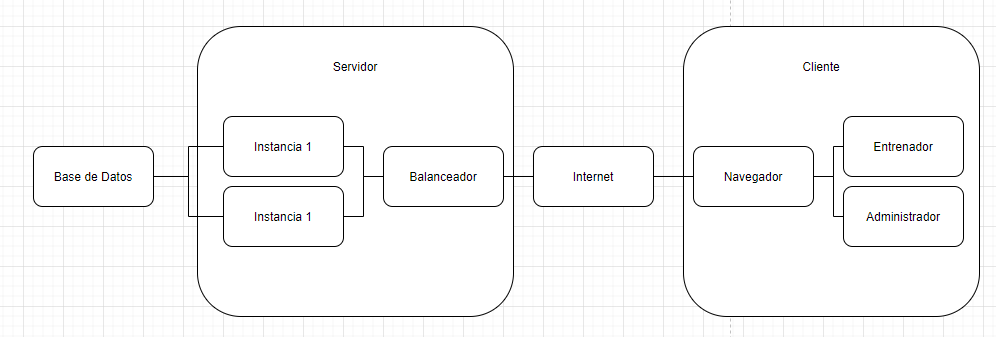


Figure 3: Diagrama de bloques

### Modelo de Proceso Empresarial

En este modelo de proceso empresarial se muestra la relación entre los componentes del software y otros externos en los que se apoya.

En la versión actual no tiene ninguna integración con otros sistemas externos, así que se ha tomado la libertad de incluir posibles funcionalidades futuras:

* Envío de progresión de una gráfica a un cliente por whatsapp.

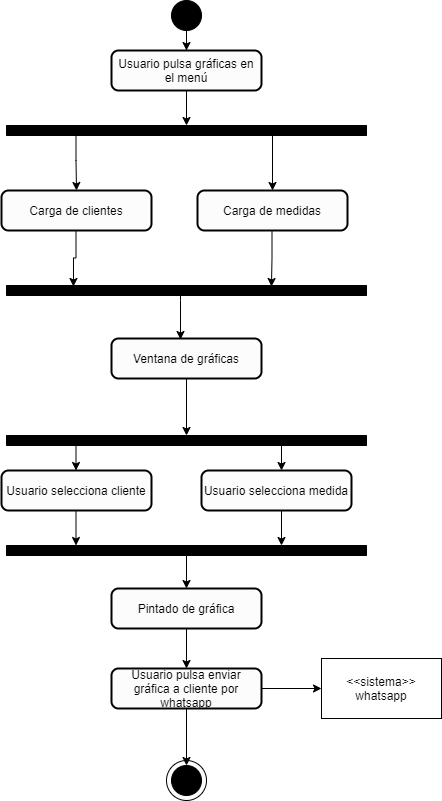


Figure 4: Modelo de proceso empresarial

## Diseño de la Arquitectura Técnica

Se pretende describir la estructura hardware que requiere este proyecto y su relación con el software desarrollado.

### Aspectos Organizativos

Actualmente únicamente hay dos actores humanos que influyen en los mismos procesos de negocio:

* Entrenador: Usuario principal
* Cliente: Cómo se ha comentado, uno de los objetivos es que el cliente no necesite utilizar la herramienta.

Todos los procesos que automatiza el software actualmente se hacen a mano o con ayuda de herramientas individuales independientes en casos puntuales.

Con la inclusión de esta herramienta aparece un nuevo actor:

* Administrador: Será el encargado de dar de alta nuevos usuarios en la aplicación y gestionar los permisos de estos.

### Tecnología hardware a utilizar

No se ha propuesto ningún hardware específico para la instalación de esta aplicación para cumplir con el objetivo de portabilidad.

Cómo requisito mínimo unicamente es necesario un servidor que pueda ejecutar un contenedor de Docker y que el cliente se conecte a través de un navegador (idealmente Google Chrome).

Un modelo hardware recomendado sería el siguiente:

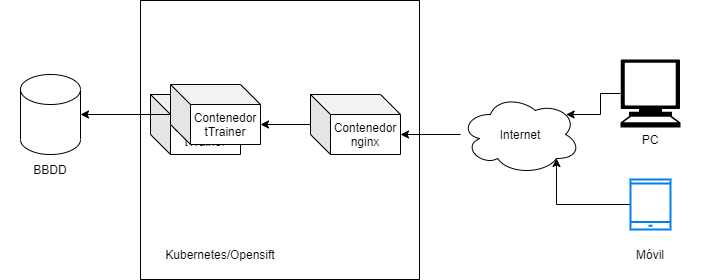


Figure 5: Diseño hardware

Instalación ideal para un entorno cloud privado o público.

* BBDD como servicio independiente, idealmente mysql aunque se podría cambiar a cualquier otro fácilmente.
* Servicio de ejecución de contenedores especializado como puede ser Kubernetes u Opensift.
  + Habría que incluir información adicional de despliegue para las instancias de la aplicación.
  + Habría que añadir a la instalación una instancia de un balanceador de carga cómo puede ser nginx.
* Varios clientes, uno web, otro móvil IOS, otro móvil Android…

### Fronteras de mecanización

Con la aplicación desarrollada hasta el punto aquí descrito, todo el trabajo del entrenador estaría informatizado, pero faltaría lo siguiente:

* Creación de usuarios: Esta lógica que actualmente realiza el Administrador, debería ser pasado a un proceso automático de registro.
* Comunicación entrenador-cliente: Actualmente se delega al 100% esa comunicación al Entrenador, en un futuro debería implementarse procesos de integración que permitan incluir esta comunicación en la app.
* Cliente: Actualmente uno de los objetivos es que los clientes no necesiten utilizar la aplicación. Aunque en un futuro se podrían incluir funcionalidades para que puedo hacerlo si lo desea, por ejemplo pudiendo ver sus progresos bajo demanda.

### Plan de Fases restantes

Si consideramos que la aplicación estaría lista para salir al mercado quedarían las siguientes Fases para una correcta puesta en Producción:

* Elección de hardware y contratación del mismo.
* Adaptación del software al hardware contratado.
* Despliegue del software.
* Pruebas de aceptación en producción.
* Fase de explotación

Aun así se considera que el software no está en un estado lo suficientemente maduro para poder salir al mercado, se necesita mucho esfuerzo adicional en nueva funcionalidad y optimización por ejemplo de la interfaz gráfica.

## Tabla de influencia de Requisitos

En este apartado se procede a organizar los requisitos en función de la capa de la arquitectura dónde se cumplen.

La explicación general de que funcionalidades se recogen en cada una de las capas es la siguiente:

|  |  |  |
| --- | --- | --- |
| **Capa** | **Funciones** | **Paquete** |
| **Presentación** | * *Navegación entre ventanas* * *Comunicación con el usuario* * *Validaciones de datos en cliente* | * *view* |
| **Lógica** | * *Lógica de negocio* * *Cálculos y procesamiento de datos* * *Control de errores* | * *service* * *util* * *infra* * *config* |
| **Datos** | * *Almacenamiento de la información* * *Modelado de la información* | * *repository* * *model* |

Table 1: Funciones de las capas de la arquitectura

Con esta explicación inicial esta es la relación de estas capas con los requisitos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificador** | **Nombre** | **Presentación** | **Lógica** | **Datos** |
| R1 | Gestión de usuarios | Ventanas gráficas | Lógica de negocio | Modelo de negocio |
| R2 | Gestión de clientes | Ventanas gráficas | Lógica de negocio | Modelo de negocio |
| R3 | Gestión de progresos de los clientes | Ventanas gráficas | Lógica de negocio | Modelo de negocio |
| R4 | Gráficas de progresión de los clientes | Ventanas gráficas |  |  |
| R5 | Gestión de calendario de usuarios | Ventanas gráficas | Lógica de negocio | Modelo de negocio |
| R6 | Gestión de menús de clientes | Ventanas gráficas | Lógica de negocio | Modelo de negocio |
| R7 | Gestión de plantillas de menús | Ventanas gráficas | Lógica de negocio | Modelo de negocio |
| R8 | Gestión de tablas de entrenamiento de clientes | Ventanas gráficas | Lógica de negocio | Modelo de negocio |
| R9 | Versión de app móvil | Programación completa en otro lenguaje | Separación del back  Exponer una interfaz en un web service |  |
| R10 | Portabilidad |  | Arquitectura compatible |  |
| R11 | Login | Ventanas gráficas | Lógica de negocio | Modelo de negocio |

Table 2: Tabla de influencia de requisitos

## Diseño de Componentes

Como ya se ha comentado con anterioridad la aplicación se ha diseñado como una arquitectura en capas, intentando seguir el patrón modelo-vista-controlador. Además sigue un modelo de aplicación web para la comunicación con los usuarios.

Esto unido a la elección de tecnologías, más concretamente JSF y spring hace que vez de separar la aplicación en componentes lógicos o por funcionalidad de negocio se dividan por capa técnica de la siguiente manera:

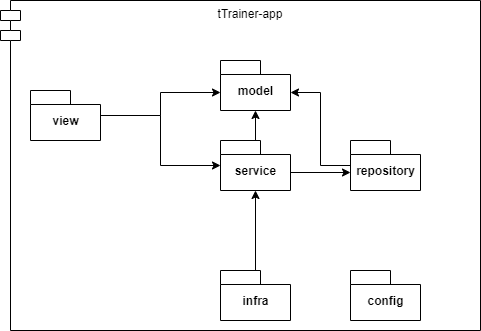


Figure 6: Diagrama de Componentes

La descripción de cada componente es la siguiente:

* model: modelo de negocio
* repository: persistencia del modelo de datos
* service: lógica de negocio
* view: controladores de la capa de vista
* config: configuraciones generales al programa
* infra: lógica encargada de la infraestructura técnica

A estos paquetes habría que añadir otros artefactos que no son código Java:

* webapp: Código xhtml con la vistas jsf de la aplicación
* resources: ficheros de propiedades de la aplicación y scripts de inicialización de la base de datos en memoria
* docker: Dockerfile para empaquetar la aplicación como un contenedor de Docker y cumplir el criterio de portabilidad.

## Diseño de Interfaz

En este apartado se define el diseño de la interfaz gráfica de la aplicación mediante un diagrama de navegabilidad y un prototipo de cómo debería ser la interfaz en su versión móvil.

### Diagrama de navegabilidad

En el siguiente diagrama se tienen en cuenta las relaciones entre las distintas vistas de las que dispone la aplicación y quedan excluidas las que son proporcionadas automáticamente por librerías externas, cómo pueden ser las páginas de error.

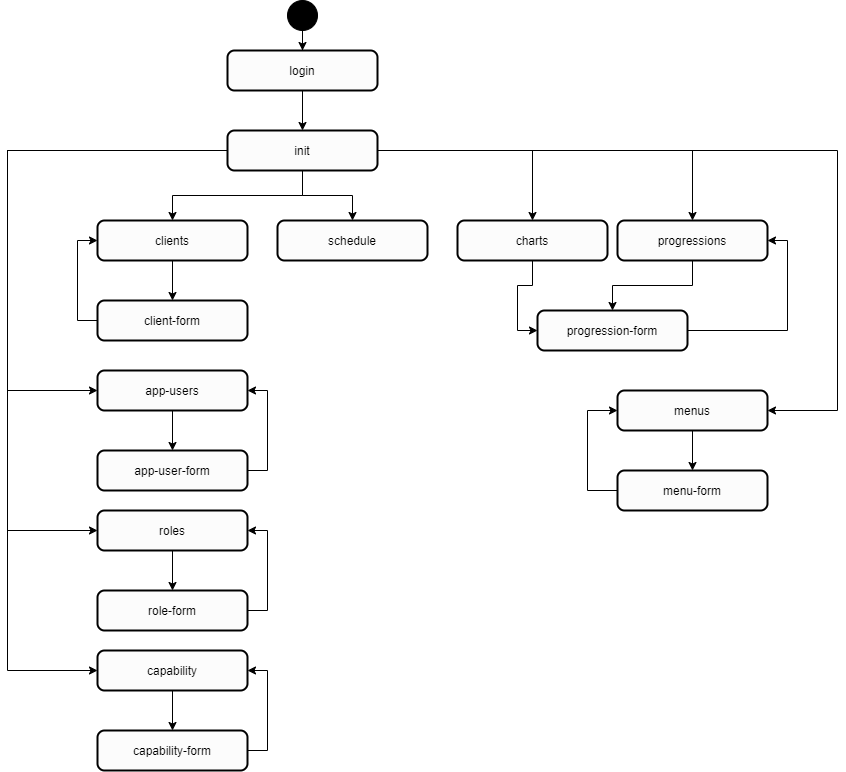


Figure 7: Diagrama de navegabilidad

### Diseño de pantalla

Aunque el diseño actual de la interfaz gráfica es plenamente funcional, en los requisitos queda claro que se debería realizar un diseño adaptado y eficiente para dispositivos móviles.

Por este motivo a continuación se procede a mostrar un prototipo de pantalla móvil, concretamente la vista de gráficas con más funcionalidad que actualmente:

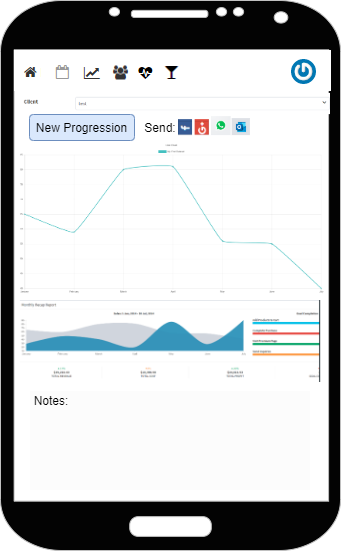


Figure 8: Diseño de pantalla

## Diseño de base datos

Con el objetivo de cumplir el requisito de portabilidad se ha intentado diseñar un esquema de base datos que sea compatible o fácilmente migrable a cualquier sistema de bas de datos actual, aunque por razones técnicas el código SQL se ha tenido que adaptar a estándares mysql.

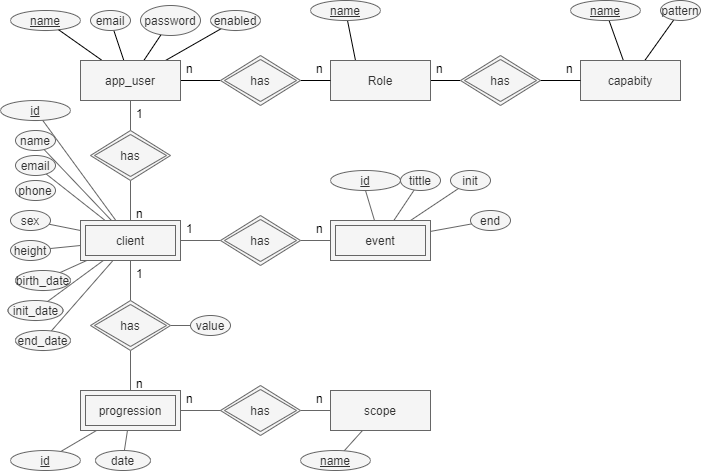


Figure 9: Diagrama entidad relación