

Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática  
Devising A Project

BanquetBuddy

## Procedimiento SCRUM



Grado en Ingeniería Informática – Ingeniería del Software  
Ingeniería del Software y Práctica Profesional

Curso 2023 – 2024

Fecha	Versión
17/02/2024	1.0

Grupo de prácticas: 8
Alberto Benitez Morales
Álvaro Carrera Bernal
Álvaro Navarro Rivera
Álvaro Jose Sanchez Flores
Artemio Rodriguez Asensio
Eduardo de Bustamante Lucena
Fernando Barroso Barroso
Francisco Jose Vargas Castro
Gonzalo Santiago Martín
Guillermo Alonso Pacheco Rodrigues
Jaime Caballero Hernandez
Javier Nunes Ruiz
Javier Rodríguez Cordero
Juan Martínez Cano
Marco Antonio Roca Rodríguez
Mario Sanchez Naranjo
Pablo Martínez Valladares

**Control de Versiones**

Fecha	Versión	Descripción
17/02/2024	1.0	Primera versión del documento

## Contenido

<b>1. Nivel de grupo general</b>	<b>4</b>
1.1 Sprint Planning (Planificación del Sprint)	4
1.2. Reunión Scrum de Scrum (Análoga a Daily Stand-Up en subgrupos).	4
1.3. Sprint Review (Revisión del Sprint)	5
1.4. [SI PRECISA]. Backlog Refinement (Refinamiento del Backlog)	5
<b>2. Nivel de subgrupos</b>	<b>6</b>
2.1. Sprint Planning (Planificación del Sprint)	6
2.2. Daily Stand-Up	6
2.3. Sprint Retrospective (Retrospectiva del Sprint)	7
2.4.[SI PRECISA]. Backlog Refinement (Refinamiento del Backlog)	7
<b>3.Requisitos de calidad de la app</b>	<b>9</b>
1º Puesto:	9
2º Puesto:	9
3º Puesto:	10
<b>2. Consideraciones generales</b>	<b>10</b>
<b>4.Posible división del equipo</b>	<b>10</b>
1º Puesto: Requisitos Básicos	10
2º Puesto: Funcionalidades Avanzadas	11
3º Puesto: Descubrimiento y Compartición	11
Consideraciones Generales para el Testing	11

# 1. Nivel de grupo general

## 1.1 Sprint Planning (Planificación del Sprint)

- **Objetivo:** Decidir qué se trabajará en el próximo Sprint.
- **Participantes:** Scrum Master de cada subgrupo + Product Owner.
- **Procedimiento:**
  - Se analizan los ítems que hay que implementar del **Backlog general**.
  - El equipo selecciona las tareas para el presente Sprint.
  - Los ítems se evalúan en función de su complejidad según estimación póquer para una distribución equitativa entre cada subgrupo.
  - Los ítems se asignan a cada subgrupo.
  - Se establece "Definition of Done" (el único, para todos los grupos)
- **Duración:** no puede superar 2 horas.
- **Momento:** antes de empezar Sprint
- **Documentación:**
  - 1) el Sprint Backlog(en una hoja de cálculo)
  - 2) Acta de reunión "Sprint Planning"
  - 3) Definition of done

## 1.2. Reunión Scrum de Scrum (Análoga a Daily Stand-Up en subgrupos).

- **Objetivo:** Sincronización entre subgrupos e identificación de impedimentos.
- **Participantes:** Scrums Masters de cada subgrupo.
- **Procedimiento:**
  - Cada miembro del equipo explica:
    - 1) Qué está hecho de momento por su subgrupo
    - 2) Qué queda por hacer

### 3) Qué impedimentos hay (si hay)

- **Duración:** 15 minutos.
- **Momento:** cada 4-5 días durante el Sprint.
- **Documentación:** Acta de reunión “Scrum de Scrum”

## 1.3. Sprint Review (Revisión del Sprint)

- **Objetivo:** Presentar el trabajo completado durante el Sprint, sincronizarse entre subgrupos.
- **Participantes:** Mínimo un representante de cada equipo para presentar el trabajo de su subgrupo y Product Owner. Idealmente, participa todo el grupo (si la reunión se lleva a cabo en Discord, los que no presentan el trabajo tienen micro silenciado y piden permiso para hablar).
- **Procedimiento:**
  - Cada Scrum Master presenta lo que ha desarrollado su equipo y recoge feedback de los participantes.
  - El Product Owner actualiza el backlog según el feedback y el progreso.
- **Duración:** No puede superar 3 horas.
- **Momento:** suele hacerse justo al final del Sprint, **En nuestro caso particular**, mejor marcar una fecha cuando el Sprint esté a 80% de completarse, para que nos dé tiempo a corregir algo y recoger feedback más completo de cada subgrupo.
- **Documentación:** Acta de “Sprint Review”

## 1.4. [SI PRECISA]. Backlog Refinement (Refinamiento del Backlog)<sup>1</sup>

- **Objetivo:** Revisar y actualizar el backlog del producto. En otras palabras, si se pretende hacer algo tan sensible como modificar algo en el Backlog, todos los subgrupos deben estar al tanto y ser conscientes del impacto que podría producir en sus tareas. Nos ahorramos disgustos y conflictos llegando a un acuerdo entre todos.
- **Participantes:** Product Owner y los Scrum Masters de cada equipo. Si no puede un Scrum Master, debería haber un representante del subgrupo.
- **Procedimiento:**
  - Revisar, detallar y estimar los ítems del backlog.

---

<sup>1</sup> Incluso durante un sprint, puede haber necesidad de ajustar el backlog del sprint. Esto puede suceder si se descubre que ciertas tareas son más complejas de lo esperado, si se completan tareas más rápidamente de lo previsto, o si surgen nuevas tareas urgentes.

- Priorizar los ítems para los próximos Sprints.
- **Duración:** no superior a 1 hora.
- **Momento:** cuando precise.
- **Documentación:**
  - 1) Backlog modificado(hoja de cálculo)
  - 2) Acta de reunión “Backlog Refinement”.

## 2. Nivel de subgrupos

### 2.1. Sprint Planning (Planificación del Sprint)

- **Objetivo:** Decidir qué se trabajará en el próximo Sprint.
- **Participantes:** Scrum Master de cada subgrupo y el Equipo de desarrollo.
- **Procedimiento:**
  - Se analizan los ítems que hay que implementar del **Backlog del subgrupo**.
  - El equipo selecciona las tareas para el presente Sprint.
  - Los ítems se evalúan en función de su complejidad según estimación póquer para una distribución equitativa entre cada miembro del subgrupo.
  - Los ítems se asignan a cada persona.
- **Duración:** no puede superar 2 horas.
- **Momento:** después de haber hecho el Sprint Planning a nivel general, antes de empezar el Sprint del subgrupo.
- **Documentación:**
  - 1) el Sprint Backlog del subgrupo (en una hoja de cálculo)
  - 2) Acta de reunión “Sprint Planning” del subgrupo

### 2.2. Daily Stand-Up

- **Objetivo:** Sincronización diaria e identificación de impedimentos.
- **Participantes:** Equipo de desarrollo y Scrum Master del subgrupo.
- **Procedimiento:**
  - Cada miembro del equipo explica:

- 1) Qué está hecho de momento
  - 2) Qué queda por hacer
  - 3) Qué impedimentos hay (si hay)
- **Duración:** 15 minutos.
  - **Momento:** Idealmente, todos los días. **Si no se puede reunión, hablar en el grupo.**
  - **Documentación:** Acta de reunión “Daily Stand-Up”

## 2.3. Sprint Retrospective (Retrospectiva del Sprint)

- **Objetivo:** Reflexionar sobre el Sprint pasado para mejorar en el siguiente.
- **Participantes:** Equipo de desarrollo, Scrum Master del subgrupo.
- **Procedimiento:**
  - Discutir qué fue bien, qué podría mejorar y cómo implementar esos cambios.
  - Crear un plan de acción para las mejoras identificadas.
- **Duración:** no superior a 3 horas.
- **Momento:** el último/penúltimo día del Sprint del subgrupo.
- **Documentación:** La retrospectiva con el análisis.

## 2.4.[SI PRECISA]. Backlog Refinement (Refinamiento del Backlog)

- **Objetivo:** Revisar y actualizar el backlog del producto. En este caso, obligatoriamente hay que hacer Backlog Refinement en el grupo general, ya que puede afectar significativamente a otros grupos los cambios en el Sprint Backlog de un subgrupo(si hay tareas de un grupo que dependen del otro).
- **Participantes:** Scrum Master del subgrupo y Equipo de desarrollo.
- **Procedimiento:**
  - Revisar, detallar y estimar los ítems del backlog.
  - Priorizar los ítems para los próximos Sprints.
- **Duración:** no superior a 2 horas.
- **Momento:** cuando precisa.
- **Documentación:** Acta de reunión con las propuestas de modificación de cambio del Backlog del Subgrupo. No se puede modificar Backlog del subgrupo, ni Backlog general

antes de la reunión Backlog Refinement del subgrupo general (ver procedimiento de Backlog Refinement del grupo general).



### 3.Requisitos de calidad de la app



Diapositiva presentación, página 10.

#### 1º Puesto:

- **Progressive:** Funciona para todos los usuarios, independientemente del navegador elegido, utilizando mejoras progresivas.
- **Responsive:** Se ajusta a cualquier tamaño de pantalla y dispositivo (móvil, tablet, portátil, etc.).
- **Faster** (after initial loading): Utiliza técnicas como el almacenamiento en caché estratégico para garantizar que la aplicación se cargue rápidamente después de la primera visita.
- **App-like Interactions:** Interacciones con el usuario fluidas y con una sensación de aplicación nativa.

#### 2º Puesto:

- **Fresh:** El contenido se actualiza constantemente.
- **Re-engageable:** La aplicación debe utilizar notificaciones push para mantener a los usuarios comprometidos.
- **Connectivity Independent:** Funciona sin conexión o en redes de baja calidad.

### 3º Puesto:

- **Discoverable:** La aplicación debe ser fácilmente identificable como una aplicación web por los motores de búsqueda.
- **Linkable:** Debe ser compartida fácilmente a través de un URL.

## 2. Consideraciones generales

- **Puesto 1:** Es la base. Una PWA necesita ser progresiva, responsiva, más rápida después de la carga inicial, tener interacciones similares a las aplicaciones nativas y ser segura.
- **Puesto 2:** Construye sobre el puesto 1, no necesariamente incluye las características del puesto 1, pero sin ellas, las funcionalidades de re-engagement y operatividad offline no serían tan efectivas.
- **Puesto 3:** Depende de los fundamentos del puesto 1 y las mejoras del puesto 2, pero se centra más en la capacidad de ser descubierto y en la facilidad para compartir la aplicación con otros usuarios.

## 4.Posible división del equipo

Podemos hacer un grupo de Frontend, 2 grupos de Backend y un grupo de testing.

La opinión del Chatgpt de qué tiene que hacer cada grupo para cumplir con los requisitos de calidad:

### 1º Puesto: Requisitos Básicos

- Progressive (**Backend y Frontend**):
  - **Backend:** Crear APIs robustas que puedan servir datos a diferentes navegadores y dispositivos.
  - **Frontend:** Implementar mejoras progresivas en la interfaz para garantizar que funcione en todos los navegadores.
- Responsive (**Frontend**):
  - **Frontend:** Diseñar y desarrollar una interfaz de usuario que se ajuste y responda a diferentes tamaños de pantalla y dispositivos.

- Faster (after initial loading) (**Backend y Frontend**):
  - **Backend**: Optimizar la entrega de datos, como implementar almacenamiento en caché en el servidor.
  - **Frontend**: Técnicas de carga perezosa, optimización de recursos como imágenes y scripts.
- App-like Interactions (**Frontend**):
  - **Frontend**: Crear una interfaz fluida con animaciones y transiciones que imiten una aplicación nativa.

## 2º Puesto: Funcionalidades Avanzadas

- Fresh (**Backend y Frontend**):
  - **Backend**: Asegurar que el servidor proporcione datos actualizados.
  - **Frontend**: Implementar estrategias para actualizar la interfaz de usuario con nuevos datos.
- Re-engageable (**Backend y Frontend**):
  - **Backend**: Implementar el soporte para notificaciones push.
  - **Frontend**: Manejar la recepción y presentación de notificaciones push.
- Connectivity Independent (**Backend y Frontend**):
  - **Backend**: Diseñar APIs que puedan funcionar eficientemente en condiciones de baja conectividad.
  - **Frontend**: Implementar estrategias de trabajo offline, como el uso de Service Workers y almacenamiento local.

## 3º Puesto: Descubrimiento y Compartición

- Discoverable (**Frontend y Backend**):
  - **Backend**: Asegurar que la aplicación provea metadatos adecuados para los motores de búsqueda.
  - **Frontend**: Optimización de SEO, como la implementación de etiquetas semánticas y metadatos.
- Linkable (**Frontend**):
  - **Frontend**: Diseñar la aplicación para que sea fácilmente compartible, asegurando que cada página o vista tenga una URL única que pueda ser compartida.

## Consideraciones Generales para el Testing

- Testing (Equipo de Pruebas):

- Probar todas las funcionalidades en diferentes navegadores y dispositivos para garantizar la progresividad y la responsividad.
- Verificar la eficiencia del almacenamiento en caché y la carga rápida después de la primera visita.
- Probar la aplicación en condiciones de red variadas para asegurar su funcionamiento en modo offline.
- Asegurarse de que las notificaciones push se entreguen y funcionen correctamente.
- Realizar pruebas de SEO para garantizar que la aplicación sea descubrible.