

Unidad 4 - Prácticas

Boards

Tras registrarnos en Azure DevOps, realiza las siguientes tareas:

Práctica 1: Creación de un proyecto Basic

- Crea una organización, con el nombre que prefieras
- Crea un proyecto en la organización creada, llamado RealStateApp

Añade si quieres una descripción al proyecto. Tendrá los siguientes requisitos:

- Privado
- Con Git como sistema de versión de controles
- Con *work item process* Basic

Explora con tranquilidad el proyecto, descubre todas las secciones que tienen y las posibilidades que ofrece. Es un proyecto de prueba, no es peligroso realizar los cambios que necesites para explorar.

Práctica 2: Creación de un proyecto Agile

En la organización ya creada, crea un proyecto, llámalo The GoodHome.

Tendrá las siguientes especificaciones:

- Privado
- Con Git como sistema de versión de controles
- Con *work item process* Agile
- Si te es posible, añade a una persona a tu proyecto para que colabore contigo en estas prácticas.

Trabajar en solitario en un proyecto no es algo extraño, pero no ocurre la mayor parte de las veces. La parte más importante del DevOps es la colaboración y comunicación constante en el equipo. Acostumbrarse a trabajar en equipo es una de las habilidades más valiosas en el mundo del desarrollo.

- Crea una wiki y añade una primera página con una descripción de la aplicación para gestión de viviendas y alquiler de inmuebles.
- Añade en la primera página el nombre de los miembros de tu equipo, así como el rol que ocupan.

La documentación es vital para un proyecto, y Markdown es una forma muy común de crear documentos en repositorios y otras localizaciones para compartir código y conocimientos asociados.

Práctica 3:

El equipo de NeonCoding te da la bienvenida por unirse al proyecto de desarrollo de TheGoodHome. Esta aplicación de gestión de inmuebles lleva mucho tiempo en el mercado, y como todo, necesita constante renovación y mantenimiento.

En el mensaje de bienvenida, te escriben sobre la situación actual del proyecto de una forma muy general. El mensaje dice así:

Tras darte la bienvenida, es hora de que conozcas la aplicación que desarrollamos y mantenemos desde hace ya 7 años. TheGoodHome es un portal para visualizar las viviendas e inmuebles en venta y alquiler, valorarlas y compartirlas con posibles compradores o arrendados.

En los últimos tiempos hemos estado experimentando problemas con algunas secciones de la aplicación. El rendimiento es mejorable, las búsquedas tardan mucho en realizarse. Existen varios filtros por los que realizar las búsquedas de viviendas, pero están incompletos. Los posibles clientes necesitan buscar por viviendas en construcción, a la venta, en alquiler. El filtro de búsqueda por alquiler no aparece en la web. Otro problema que nos ocupa es que las valoraciones de las viviendas no corresponden a los valores que se almacenan en la base de datos.

La última entrega a cliente nos llevó mucho tiempo, las pruebas están consumiendo mucho tiempo y tenemos constancia de que no se están realizando todas las que deberíamos.

Incluso en algunas ocasiones aparecen elementos que no corresponden a las búsquedas realizadas, aparecen alquileres en viviendas a la venta.

Esperamos que te acomodes pronto al desarrollo de TheGoodHome, mucho éxito y *happy coding*.

Tras este mensaje, revisas Boards y ves que ninguno de estos problemas que comentan en el mensaje de bienvenida no están reflejados como *work items* en el Backlog o en el Work Items.

Es hora de ponerle remedio a eso. Notificando a la líder del equipo, Alex, empiezas a crear los *work items* que hacen falta.

Tu tarea es:

- Analiza el mensaje
- Extrae los *work items* que detectes
 - Agrupa los problemas por User Stories
 - Crea Tasks para las User Stories que has creado

Práctica 4

La práctica anterior debe haber producido varias User Stories y también Tasks asociadas a ellas. Es hora de organizar este trabajo pendiente para abordarlo poco a poco, a través del tiempo.

La primera semana en NeonCoding, trabajando para TheGoodHome, será de documentación e investigación. Por lo que empezarás a resolver problemas la semana que viene, pronto estarás programando para la aplicación.

Tu tarea es:

- Crea un nuevo sprint para la semana siguiente de la que te encuentres, llámalo "Sprint_Inicial", con dos semanas de duración.
- Asocia todos estos *work items* recién creados al nuevo sprint
- Visita el sprint creado y distribuye los *work items* que has creado a lo largo de las dos semanas que dura el sprint.

Práctica 5

Observas los *work items* que tienes creados y piensas que es mucho trabajo para una sola persona, el resto del equipo debería estar contigo

- Invita a más personas a tu equipo, utiliza los siguientes contactos:
 - Yeray: yeray@codehouse.academy
 - Alex: alex@codehouse.academy
 - Pau: pau@codehouse.academy
- Asigna tareas a los miembros del equipo recién llegados

Una de estas nuevas incorporaciones es Yeray, un programador junior con apenas dos meses de experiencia. Su capacidad de trabajo es de seis horas, mientras que el resto tiene una capacidad de siete horas.

Por otra parte, Alex tiene un día de vacaciones: el primer viernes de la primera semana del sprint, por lo que debe quedar reflejada su ausencia en el sprint. Pau también está de vacaciones, pero solo el lunes y el martes de la segunda semana del sprint. Por lo demás, todo el equipo estará presente en las dos semanas de desarrollo del sprint.

- Asigna las capacidades adecuadas a los miembros del equipo.
- Refleja en el sprint los días de ausencia del equipo.

Repos

Práctica 1:

Es hora de empezar a trabajar con el código que tenemos. Ahora mismo, el código con el que estamos trabajando se encuentra en un repositorio de CodeHouse Academy. A partir de este trabajarás y tendrás tu propia copia.

- Clona en tu equipo el [repositorio de CodeHouse](#), que se encuentra en la URL indicada.
- Cambia el remote para subirlo al repositorio de tu propio proyecto.
- Sube tu repositorio local al repositorio del proyecto

Ahora tienes tu propia copia del repositorio en tu proyecto, y puedes empezar a trabajar con ese código.

Práctica 2

Vas a realizar el primer cambio sobre el repositorio recién creado, y por supuesto debe tener su propia *branch*. Para ello, crearás una nueva *branch*, la descargarás en tu repositorio local, en tu equipo.

Ahora trabaja sobre esa *branch* y crea un *commit* en esta, para subirla al repositorio de tu proyecto.

Tu tarea será:

- Crea una nueva *branch* llamada *feature/cambios-iniciales*
- Realiza una modificación, realiza un *commit* y subelo al repositorio
- Revisa en el repositorio que la *branch* existe y también el *commit* recién creado.

Práctica 3

Los cambios que has realizado deben ser incorporados a la *branch* principal, por lo que es necesario hacer que estos cambios pasen por una validación y revisión por parte del equipo. Esto lo hacemos a través de *pull requests*.

Lo que debes hacer es:

- Crea un pull request y:
 - Asocia un work item ya creado en tu proyecto
 - Añade a otra persona como *reviewer*
- Comenta los cambios que aparecen, en la sección de *discussion*.
- Revisa el código y añade un comentario en una de las líneas del código modificado

Práctica 4

Este último cambio introducido en el *pull request* de la práctica anterior era lo necesario para marcar un punto de inflexión en la aplicación. El código que hemos creado es suficiente como para crear una *tag*, una etiqueta de versión, en la rama *master*.

Tu tarea será:

- Crea una *tag* a partir del último *commit* realizado sobre la *branch* *master*
- Dale un nombre, *v1*, describiendo también los cambios que incluye
- Revisa que sea la única *tag* existente actualmente

Práctica 5

Las *branches* con mucho tiempo de vida tienen código desactualizado y eventualmente deben ser destruidas. La rama *features/cambios-iniciales* ha cumplido su función, por lo que ya puede eliminarse.

- Elimina la rama *features/cambios-iniciales*

Las ramas eliminadas no se pueden recuperar, no elimines las ramas principales o las que estén activas.